

# PROYECTO DE FIN DE CARRERA

Este proyecto analiza las capacidades de un sistema de verificación de identidad de un usuario accediendo mediante la lectura de un código QR, utilizando un proveedor de identidad descentralizado basado en OpenID.

Se realiza la implementación de un escenario modelo describiendo su desarrollo técnico y por último se analizan los resultados obtenidos

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

UNIVERSIDAD CARLOS III DE MADRID



Autor **AINGERU MEDRANO ISLA**

Tutor **DANIEL DÍAZ SÁNCHEZ**

Leganés. Agosto de 2013

*Análisis e  
implementación de  
sistema de  
verificación de  
identidad mediante  
acceso por código  
QR*

# Resumen

---

Este estudio como proyecto de fin de carrera, propone la implementación de un escenario modelo involucrando dos conceptos de interés técnico, los servicios de gestión de identidad y la codificación QR. Existen importantes ventajas en el uso de este servicio, sobre todo en lo referente a la identidad digital, vista como un conjunto de atributos que pueden ser solicitados de forma independiente. Trabajaremos con códigos QR en un entorno poco explorado, tratando de integrar las comunicaciones máquina a máquina entre un proveedor de servicio y un proveedor de identidad. Durante el desarrollo de esta memoria se describirá la implementación del escenario tanto a nivel técnico, como funcional y se analizarán los resultados obtenidos dentro del contexto marcado por el propósito del estudio.

En primer lugar se hará mención a los objetivos que se deben cumplir con el estudio, así como la motivación que promueve este desarrollo. Se hará un repaso al contexto establecido por las tecnologías y herramientas que se utilizarán. Esto permite que comprendamos bien los conceptos que se manejan a lo largo del estudio.

Gran parte de esta memoria está dedicada al desarrollo técnico sobre la implementación del escenario. Dicho desarrollo plantea incertidumbres en sus comienzos y muchos retos que se describen detalladamente, así como las decisiones tomadas al respecto. Se aplican técnicas y conocimientos en materia de Software durante todo el desarrollo que permiten finalizar con éxito el escenario.

Por último se analizan los resultados arrojados por el escenario y los pasos seguidos para la implementación. De este análisis, obtenemos conclusiones e interesantes ideas de aplicación alternativas que nos permitirían extender aún más este trabajo.

Se dedica un importante espacio en esta memoria para las apreciaciones del alumno, su visión del trabajo realizado, teniendo en cuenta su trayectoria personal. También se incluye la historia del proyecto desde la perspectiva del alumno, que de alguna forma refleja el esfuerzo realizado en términos más particulares.

# Abstract

---

This study as Degree Thesis, suggests a sample environment implementation, introducing two interesting concepts, user account management services and QR coding. There are a lot of advantages using these services, about digital identity in special, seen as a group of attributes that can be requested independently. We will work with QR codes in an unexplored environment, trying to integrate machine to machine communications between a service provider and an identity provider. Along this report, we will describe the environment implementation, in tech terms and functional terms. We will also analyze our results in the study context.

First, we will define the targets that we need to fulfill in order to success, and the motivation that suggests this development. We will make a reminder about de technologies and tools in our context. This will make the concepts used by this study, more comprehensive for us.

A great part of this report is written about the technical development. This development raises some questions at first and a lot of challenges that will be described in great detail. Also the chosen answers will be described in detail. A lot of techniques and knowledge about Software Science are applied in order to accomplish our goal.

Finally, we will analyze our environment results, and the tasks that we make implementing it. By this analysis, we also get some conclusions and interesting ideas that could let us extend our study for alternative applications.

A great space is dedicated for the student suggestions, his vision about the study, keeping in hand his personal projection. Of course, it includes the Thesis story by the student perspective, so its shows the efforts in a different manner.

# Agradecimientos

---

A mis padres y a mi hermana.

A mi mujer y a mi otra familia, gracias María.

A mis amigos, uno a uno.

A Dirasoft, por darme tanto y quitarme tanto.

A todos los profesores que he conocido en mi vida, incluyendo a Dani.

Os debo prácticamente todo lo que sé y lo que soy.

*Todos los que alguna vez y de alguna forma, me han animado a que haga  
este pequeño gran esfuerzo.*

*Sin vuestra profunda insistencia no hubiese sido posible.*

# Índice de contenidos

---

1.	Introducción y objetivos.....	7
1.1.	Motivación del proyecto .....	7
1.2.	Objetivos .....	8
1.3.	Contenido de la memoria.....	9
2.	Estado del arte .....	11
2.1.	Introducción .....	11
2.2.	Identificación automática.....	12
2.3.	Proveedores de identidad .....	15
2.4.	Comunicaciones máquina a máquina .....	18
3.	Descripción del sistema.....	21
3.1.	Arquitectura del sistema .....	21
3.2.	Descripción funcional .....	23
3.3.	Casos de uso.....	25
4.	Implementación del sistema .....	27
4.1.	Petición OpenID .....	27
4.2.	Componente de codificación QR.....	31
4.3.	Atención de peticiones HTTP .....	34
4.4.	Integración de funcionalidades .....	37
5.	Pruebas.....	42
5.1.	Pruebas de integración .....	42
5.2.	Pruebas de sistema .....	45
5.3.	Casos de prueba .....	46
5.4.	Conclusiones.....	48
6.	Historia del proyecto.....	51
6.1.	Contexto del alumno.....	51
6.2.	Historia del proyecto.....	52
6.3.	Opinión personal.....	53
7.	Conclusiones.....	55
7.1.	Análisis de resultados.....	55
7.2.	Riesgos del sistema .....	56

---

7.3.	Conclusiones generales.....	57
7.4.	Trabajo futuro .....	58
A.	Presupuesto .....	62
1.	Costes de personal .....	62
2.	Costes de material.....	63
3.	Presupuesto completo .....	65
B.	Glosario de términos.....	66
C.	Referencias.....	74
1.	Microsoft Developer Network .....	74
2.	Google Developers .....	74
3.	OpenID .....	74
4.	StackOverFlow.....	75
5.	Wikipedia.....	75
6.	Otros.....	75

# Índice de figuras

---

Figura 1: Muestras de códigos de barras en diferentes codificaciones .....	14
Figura 2: Diagrama de secuencia de un protocolo estándar de autenticación.....	17
Figura 3: Gráfico de muestra con la utilización de cuentas OpenID .....	18
Figura 4: Diagrama de capas OpenMTC .....	20
Figura 5: Esquema funcional del escenario propuesto .....	22
Figura 6: Esquema técnico del escenario propuesto .....	23
Figura 7: Estructura del proyecto en el entorno de desarrollo.....	38
Figura 8: Aspecto visual de la aplicación en tiempo de ejecución .....	41
Figura 9: Tabla de casos de prueba planteados en el escenario.....	47
Figura 10: Tabla de categorización de los casos de prueba .....	48
Figura 11: Resultados de las pruebas de integración en el entorno de desarrollo.....	49
Figura 12: Resultado correcto en la aplicación en tiempo de ejecución .....	50
Figura 13: Resultado incorrecto en la aplicación en tiempo de ejecución .....	50

# Capítulo 1

---

## 1. Introducción y objetivos

En primer lugar definiremos el propósito de este estudio así como los objetivos que se deben cumplir para que finalice de forma satisfactoria.

### 1.1.Motivación del proyecto

El proyecto surge de introducir una relación entre dos tecnologías que no se ha visto hasta ahora de forma clara. La idea de esta relación es interesante tecnológicamente puesto que abre la puerta a posibles aplicaciones en el ámbito de las tecnologías de internet y la identificación automática.

El primer concepto involucrado es conocido como la verificación de identidad descentralizada. En este concepto está basado el estándar de facto OpenID, un servicio que como veremos más adelante, permite verificar la identidad de un usuario utilizando la cuenta de un proveedor de dicho servicio, evitando que un usuario tenga que crear una cuenta personal para un determinado sitio Web, y al sitio en cuestión, la necesidad de gestionar las cuentas de usuario. Una consecuencia derivada de ello, es que los consumidores del servicio especifican exclusivamente que valores o atributos desean conocer de la cuenta de usuario, protegiendo así la privacidad del usuario de una forma que los servicios de autenticación centralizada no pueden, al estar en posesión de toda la información del usuario. Este servicio, no está muy extendido actualmente pero presenta respuestas a algunos de los problemas más habituales hoy en día para los usuarios de internet en este aspecto.

El segundo concepto es la identificación automática. Aunque el término no resulta familiar en el ámbito doméstico, los códigos de barras son conocidos en todo el mundo y engloban una gran familia de tecnologías que permiten a todo tipo de sistemas, identificar los elementos con los que trabajan. En nuestro caso, una de estas tecnologías nos resulta de especial interés debido al auge que ha tenido estos últimos años y el uso que ha recibido, esta es la codificación QR. En la actualidad esta codificación se utiliza para almacenar enlaces a contenidos de internet que pueden ser decodificados por cualquier Smartphone y acceder a dicho enlace desde el navegador del propio dispositivo. Esto es realmente interesante puesto que permite poner al alcance de cualquiera un contenido de internet desde cualquier lugar en el que se pueda colocar una imagen con un.

Estos dos conceptos ya nos dan alguna pista de la interesante relación que pueden tener entre sí. Uniendo verificación de identidad e identificación automática podemos entender el



concepto resultado como verificación de identidad automática, que es la motivación fundamental de este proyecto. En concreto, sería la verificación de identidad automática descentralizada. Cualquiera podría poner al alcance de los usuarios no solo un contenido, sino un servicio que exija la verificación de la identidad del usuario, y además valiéndonos de los estándares de verificación de cuentas ya existentes, evitando que el servicio en cuestión sea responsable de la gestión de cuentas. Esta verificación de identidad permite a efectos prácticos, que un usuario pueda acceder a servicios personales de internet como, compra electrónica, redes sociales, etc. de una forma extremadamente sencilla, a través de su Smartphone leyendo un código QR situado en cualquier lugar, como por ejemplo propagandas, televisión o cartelería.

Para plasmar esta idea y establecer su viabilidad técnica, resulta imprescindible definir un escenario práctico e implementar todas las partes necesarias. Dicho escenario de ejemplo se plantea como una máquina de Vending que requiere una verificación de edad para servir los pedidos, los cuales se realizan leyendo el código QR de la máquina. La secuencia del proceso general sería el siguiente.

- La máquina muestra un código QR con el identificador de operación que puede procesar.
- El usuario realiza la lectura del código con su Smartphone, a través de una aplicación estándar de lectura de códigos QR.
- El código leído dirige al usuario al proveedor de identidad OpenID, el cual solicita las credenciales de la cuenta.
- Introducidas las credenciales por el usuario, el proveedor re direcciona al usuario a la máquina de Vending con los datos de la cuenta solicitados.
- Si la máquina confirma que el identificador de operación es correcto y la edad establecida para el usuario es válida muestra un mensaje de operación correcta en pantalla y en el navegador del usuario.

De aquí en adelante se desarrollarán todos los esfuerzos para implementar el escenario propuesto y salvar todas las dificultades técnicas que puedan surgir. Es posible que se produzcan variaciones en el escenario propuesto debido a las limitaciones de la tecnología o modificaciones apropiadas que mejoren las capacidades del sistema. En cualquier caso, todas ellas quedarán debidamente reflejadas en las conclusiones, junto a su correspondiente justificación.

## 1.2.Objetivos

El objetivo general del proyecto es analizar las capacidades del sistema descrito, demostrar la viabilidad técnica aplicada a posibles ideas de negocio, e implementar el sistema completo para someterlo a todas las pruebas necesarias y plantear posibles líneas de estudio.

Aunque la definición inicial del sistema no es compleja, los factores que intervienen en la integración de las diferentes tecnologías y servicios que lo componen no es el todo clara, con lo que la implementación del escenario, plantea ciertos retos técnicos y arrojará conclusiones importantes que ayudarán a cumplir con el objetivo de analizar el sistema diseñado. Una vez implementado un escenario completo, será posible llevar a cabo todas las pruebas definidas. Esto nos permitirá sacar las conclusiones finales sobre la utilidad y la viabilidad de su aplicación e entornos reales de negocio.

Puesto que la implementación de un escenario para el sistema es vital para alcanzar nuestro objetivo, a continuación se enumeran las etapas generales que nos servirán para trazar y medir el avance del desarrollo.

- Estudio sobre servicios de verificación de identidad y proveedores de dicho servicio. Conseguir una perspectiva clara de su finalidad, como acceder y consumir estos servicios y a través de que proveedores.
- Estudio sobre las tecnologías de identificación automática, centradas en el estándar de codificación QR. Establecer el marco en el que se encuentra esta tecnología y completarlo con las tendencias de su uso en la actualidad.
- Definición técnica del sistema, describiendo las partes involucradas así como su comportamiento y los casos de uso.
- Definición de las pruebas a realizar, necesarias para establecer el éxito de la implementación del sistema y dar una perspectiva basada en objetivos del diseño funcional.
- Implementación del sistema software que integrará los conceptos planteados según el escenario propuesto y llevar a la práctica el sistema definido.
- Realización de las pruebas definidas. Chequeo de los resultados obtenidos, anotación de conclusiones y otras observaciones de interés.

### 1.3.Contenido de la memoria

Con el fin de organizar en términos generales los contenidos de esta memoria, y anticipar al lector un breve resumen, a continuación se describen los capítulos que componen la memoria del proyecto.

- Capítulo 1. Introducción al proyecto planteado y familiarización con los conceptos que se van a manejar a lo largo de todo el desarrollo del mismo.
- Capítulo 2. Estado del arte en el que se desenvuelve el proyecto y marco general de tecnologías involucradas.
- Capítulo 3. Descripción y definición del sistema. Diseño del escenario de prueba y planteamiento de los casos de uso para la valoración de los resultados.
- Capítulo 4. Implementación del sistema. Llevar a la práctica el escenario diseñado, explicando las dificultades encontradas y justificando las decisiones tomadas.
- Capítulo 5. Pruebas del sistema. Ejecución de las pruebas diseñadas y valoración de los resultados generales obtenidos.
- Capítulo 6. Análisis de los resultados obtenidos sobre el escenario de prueba desde distintos puntos de vista, a fin de estudiar mejoras, viabilidades u otras líneas de investigación.
- Capítulo 7. Historia del proyecto en el contexto del alumno, línea temporal del proyecto e impacto en la vida personal.

# Capítulo 2

## 2. Estado del arte

Para completar las motivaciones y propósitos del proyecto, se hará un breve repaso sobre el contexto e historia de las tecnologías involucradas en el estudio.

### 2.1.Introducción

En la actualidad podemos encontrar códigos QR en multitud de lugares, desde folletos de propaganda hasta en programas de televisión, y su popularidad crece cada día. El éxito de esta tecnología reside en la rápida respuesta para la lectura y sobre todo en el hecho de que sea posible interpretar dichos códigos QR como si se tratasen de imágenes digitales.

Sin embargo la codificación QR dispara su éxito en el momento en el que los Smartphones empiezan a incorporar cámaras de alta resolución y aparecen las primeras aplicaciones que permiten decodificar códigos de barras. Esto es todo un acontecimiento, puesto que permite a una gran cantidad de usuarios acceder al mundo de la identificación automática, un concepto muy exclusivo de ciertos sectores profesionales. Es aquí donde la codificación QR cobra sentido, ya que a través de la cámara del Smartphone puede ser decodificado como imagen digital y gana la partida a todas sus codificaciones hermanas.

La principal ventaja de las codificaciones bidimensionales frente a las unidimensionales como EAN13 o Code128 es que permiten almacenar una mayor cantidad de información. Desde hace más de cuarenta años los códigos de barras se han utilizado por todo el mundo para almacenar códigos identificadores o números de serie que servían a las compañías para relacionar de forma automática la información dentro de sus sistemas, pero el hecho de poder codificar un bloque entero de caracteres o una pequeña imagen en un código bidimensional, abre muchas puertas a otras tecnologías que pueden servirse de ello para extraer la información codificada, a través de dispositivos electrónicos y no a través de intervención humana.



Con el auge masivo de Smartphones y la proliferación de las aplicaciones de lectura de códigos, multitud de empresas lo vieron claro y los códigos QR se convirtieron en una

poderosa herramienta de marketing para atraer a los usuarios hasta sus ofertas y promociones. La culminación de esta idea se produce en el momento en el que se populariza la codificación de URLs, enlaces a contenidos de internet, de forma que un usuario de Smartphone, pulsa en la pantalla para realizar la lectura del código y automáticamente es dirigido al contenido de internet en el navegador de su Smartphone. A partir de aquí, las compañías comienzan a utilizar esta vía en masa para promocionar sus productos o sus servicios. Incluso los presupuestos del estado se han podido consultar con este sistema mientras eran presentados en tele noticiario.

Este cambio de tendencia de las compañías para captar potenciales clientes y de los usuarios para acceder a sus promociones, como cabe esperar tampoco es el final. Como muchos sabemos, una URL no tiene por qué ser un mero enlace a un contenido estático, si no que puede contener parámetros que pueden ser utilizados por el servidor receptor, como por ejemplo la ubicación GPS conocida del código QR, para realizar tareas mucho más sofisticadas que presentar un simple contenido Web. Aunque se están empezando a explotar estas características, una de las tareas más habituales en Internet que es identificarse como usuario no se le está prestando mucha atención. Que al realizar la lectura del código QR accedamos al servicio en cuestión con una identidad establecida, nos permitiría realizar compras de una forma muy sencilla, valoraciones en redes sociales o servicios de fichaje en el sector profesional, son solo algunas de las innumerables aplicaciones que se podrían alcanzar.

El objetivo que propone esta línea de estudio arroja unos beneficios más que interesantes, pero la motivación más importante para seguirla es que no se rompe la tendencia actual, en el sentido de que, el estudio no propone cambiar a los usuarios o a las compañías la forma en la que se desenvuelven. Alcanzar el objetivo obedece al estudio de encontrar la información concreta que debe codificarse dentro del código QR, de tal forma que aprovechándonos de los servicios de verificación de identidad y de todas las aplicaciones de lectura de códigos ya existentes, el procedimiento difiera poco o nada.

El estudio se desenvuelve en el contexto de las comunicaciones máquina a máquina, puesto que existe un intercambio de información entre el servicio al que accede al usuario y el proveedor de identidad. De hecho, al ser un proceso transparente para el usuario, es precisamente esta comunicación M2M la que será objeto de la investigación, y su correcta adecuación la que debe permitirnos alcanzar los objetivos marcados.

## 2.2. Identificación automática<sup>1</sup>

Así es como se conoce profesionalmente este sector, que mueve al año unos 30 millones de euros aproximadamente en España y tiene como principal objetivo automatizar y optimizar todo tipo de procesos en empresas de una gran variedad de sectores, pero especialmente las

---

<sup>1</sup> <http://www.wikipedia.org>

de corte industrial. Desde la pistola de códigos de barras en los mostradores de pequeños comercios, hasta los enormes escáneres de las líneas automáticas de paquetería, estos sistemas de identificación son críticos para muchas empresas, permitiendo realizar ciertas tareas mucho más eficazmente que mediante intervención humana, más rápido y con menor tasa de fallo.

En 1948 el presidente de la cadena local de alimentación de Philadelphia propuso a Bernard Silver, un estudiante graduado del Instituto de la Tecnología de Drexel, la investigación de algún tipo de sistema, capaz de leer la información de los productos durante su proceso de expedición. Bernard Silver y su compañero Norman Joseph Woodland trabajaron sobre diversos diseños convencidos de que era perfectamente posible desarrollar dicho sistema. Algunos de los diseños incluían elementos que no terminaron de encajar, como luz ultravioleta, debido a que eran frágiles para su manipulación o demasiado caros.

En 1949 Bernard y Norman publicaron su sistema de identificación automática, con las especificaciones completas de los dispositivos necesarios y los métodos para la codificación de la información, basada en barras y en círculos, todo ello inspirado en el código Morse. La patente se aplicó en 1952, mientras que Norman continuó su investigación en IBM, empresa que estaba especialmente interesada en ella, llegando incluso a intentar adquirir la patente sin éxito. A partir de entonces este sistema comienza a explotarse con grandísimo éxito por empresas de todo el mundo.

En 1994 la compañía Denso Wave inventa el código QR, inicialmente pensado para marcar los componentes de Toyota y mantener la trazabilidad dentro de su sistema a mayor velocidad y con más información. Sin embargo como ya sabemos, se vuelve extremadamente popular con otras finalidades.

Esta tecnología en general, que nace hace unos 60 años, y es fundamental para empresas de todo el mundo, sería un lujo exclusivo de ciertos sectores de no ser por su bajo coste. Los equipos de lectura de códigos de barras de hoy en día, aunque varían según su categoría, son asequibles para cualquier mediana o pequeña empresa, pero sobre todo destaca el bajísimo coste de las etiquetas de códigos de barras, así como las impresoras y los consumibles necesarios. Toda compañía que decide aplicar lecturas por códigos de barras para sus productos o activos se plantea cual es el coste del etiquetado en relación al producto, que en este caso es tan solo unas décimas de céntimo, lo cual no se cumple para otras tecnologías de identificación. El éxito general de esta tecnología obedece al hecho de que tiene como objetivo optimizar procesos, reduciendo costes y fallos, lo cual hace especialmente bien y asegura el retorno de la inversión de su implantación a la empresa. Si pensamos en una cadena de paquetería, hoy en día es difícil encontrar operarios ordenando y organizando paquetes en la cadena, diversos dispositivos de identificación trabajando conjuntamente con los automatismos de la cadena, evitan en este caso el coste de un buen puñado de operarios, que incluso tendrían más tasa de fallo dentro del sistema

En el siguiente cuadro, extraído del sitio Web de una empresa<sup>2</sup> privada dedicada a la integración de Software con sistemas de identificación por códigos de barras, de se muestran distintos tipos de codificaciones.



Figura 1: Muestras de códigos de barras en diferentes codificaciones

Como cabe esperar, las empresas llevan décadas trabajando sobre sistemas basados en codificaciones unidimensionales y aunque las codificaciones bidimensionales son estrictamente superiores, teniendo en cuenta que permiten almacenar mayor cantidad de

<sup>2</sup> <http://dynamicstatements.com>

información e incluso ocupan menos espacio físico, llevará mucho tiempo a las empresas de todo el mundo migrar o evolucionar sus sistemas para implantar este tipo de codificaciones. Puede incluso que los beneficios que pueda aportar esta evolución no compense la inversión en la mejora, con lo que no se podría hablar de codificaciones bidimensionales como sustitutivo de las unidimensionales. De hecho, Se especula que los códigos de barras unidimensionales existirán durante al menos otros 50 años más.

### 2.3.Proveedores de identidad<sup>3</sup>

Tras la meteórica aparición de internet en los hogares y su incesante expansión, los servicios orientados a usuario no han dejado de popularizarse, de forma que cualquiera interesado en los contenidos de un determinado sitio Web puede registrarse como usuario del sitio y acceder a contenidos extra y otras funcionalidades como noticias, compra online, participación en foros, etc.



La compra online ha sido una de las mayores revoluciones en el contexto de internet, desde el punto de vista del volumen de negocio que genera. Las tiendas de barrio quedaron atrás al aparecer las grandes superficies, lo cual supuso un gran cambio en las bases del comercio, pero a día de hoy las tiendas virtuales nos ofrecen todo su catálogo desde casa, revisiones de usuarios, videos, históricos de compra, suscripciones a noticias y servicios de entrega a domicilio entre otros. No obstante para acceder a este tipo de servicios es imprescindible registrarse como usuario, para realizar el proceso de facturación y especificar los datos de envío entre otros.

Los servicios de compra online son solo un subconjunto de servicios que exigen la identificación del usuario, pero podemos encontrar todo tipo de sitios Web incluyendo proveedores de correo electrónico, portales noticiarios o redes sociales, para los cuales es imprescindible registrarse e identificarse como usuario. Y así es como podemos ver el problema tan sencillo que se produce, un usuario de internet interesado en un buen puñado de sitios Web puede verse abrumado por la necesidad de recordar de alguna manera los datos de acceso de su cuenta de usuario para cada uno de los servicios a los que está suscrito.

Actualmente existen ciertos sistemas o aplicaciones diseñadas para paliar este problema, pero no terminan de solucionarlo, puesto que consisten en simples almacenes de cuentas y datos de acceso, no integrados con los servicios de autenticación propietarios, con lo que resultan algo molestos y en muchos casos incluso suelen ser de pago. La verdadera solución proviene del concepto de proveedor de identidad, el cual engloba el acceso a cualquier servicio que soporte dicho proveedor, si un sitio Web integra con un determinado proveedor y

---

<sup>3</sup> <http://www.wikipedia.org>



el usuario posee una cuenta en ese proveedor, no será necesario registrar una nueva cuenta y añadir nuevos datos de acceso a su posible y seguramente extensa lista de identidades.

Las grandes corporaciones como Google, Microsoft y Apple realmente actúan como proveedores de identidad de su extensa lista de productos y servicios. Asumiendo que casi la totalidad de usuarios de internet disponen de cuenta en dichos proveedores, algunos sitios Web integran con ellos asegurándose de que los usuarios accedan con mayor facilidad al no tener que registrarse y pudiendo identificarse con su cuenta de GMail por ejemplo. De hecho cada vez es más habitual entrar en sitios que permiten identificarse con la cuenta de Facebook.

OpenID surge en 2005 como protocolo estándar libre de identificación digital descentralizada, creado por Brad Fitzpatrick, con el que un usuario puede identificarse en un determinado sitio Web y verificado por cualquier servidor que soporte el protocolo. En los sitios que soportan OpenID, los usuarios no tienen que crearse una nueva cuenta de usuario para obtener acceso. En su lugar, solo necesitan disponer de un identificador creado en un servidor proveedor del protocolo OpenID, llamado proveedor de identidad o IdP. A diferencia de arquitecturas Single Sign On, OpenID no especifica el mecanismo de autenticación, la seguridad de un acceso OpenID depende de la confianza que tenga el cliente OpenID en el proveedor de identidad. Si no existe confianza en el proveedor, la autenticación no será adecuada por ejemplo para servicios bancarios o transacciones de comercio electrónico. Se habla por tanto de autenticación federada, en el sentido de que el proceso de autenticación, es delegado a un sistema independiente y de hecho OpenID es solo un estándar de los definidos en la familia de protocolos de autenticación federada.

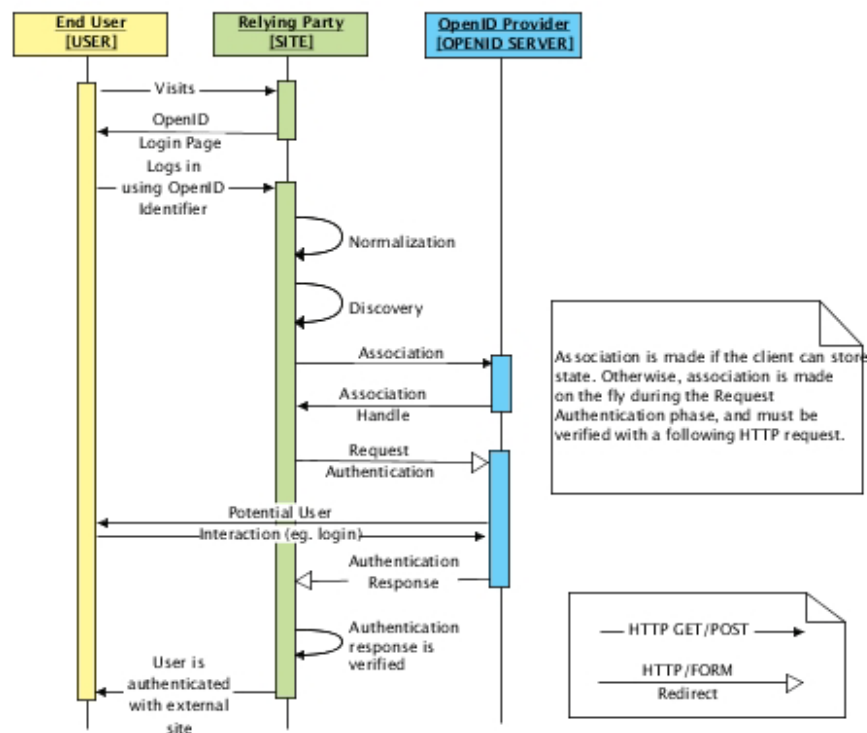


Figura 2: Diagrama de secuencia de un protocolo estándar de autenticación

La identidad digital es un concepto al que estamos habituados, y que de alguna manera define el perfil que un usuario utiliza para acceder a un sistema. En el contexto de las autenticaciones federadas, una identidad de usuario se define como un conjunto de atributos y su correspondiente valor. De esta manera una identidad puede quedar representada solamente con el subconjunto necesario de atributos requeridos por el sistema que solicita la autenticación y necesita para operar. Esta es otra gran ventaja de este servicio ya que no se requiere toda la información de la identidad digital, solo la especificada.

Es conveniente hacer una breve aclaración de conceptos, entre autenticación y autorización, ya que es habitual confundir los términos y en algunos casos incluso no diferenciarlos. La autenticación, que es el proceso con el que trabajaremos, hace mención a la verificación de la identidad de un usuario. Hablamos de un usuario autenticado cuando su identidad ha sido verificada, el usuario es realmente quien dice ser, su identidad es auténtica. Mientras que la autorización hace referencia al permiso que recibe el usuario sobre los recursos del sistema al que accede. Un usuario está autorizado a acceder a ciertos recursos o funciones de un sistema dependiendo de factores concretos definidos por el propio sistema, pudiendo estar relacionados con su identidad. En algunos casos pueden estar solapados, un usuario que no está autenticado generalmente tampoco estará autorizado, pero son en efecto dos conceptos muy diferentes.

La utilización de servicios de autenticación federada ofrece dos importantes ventajas, que ya se han mencionado. La primera es la reutilización de este servicio, ahorrando tiempos de

desarrollo en arquitectura SSO, mejorando la escalabilidad y facilitando a los usuarios la entrada a nuestro sistema. La segunda es la mejora de la privacidad sobre la identidad digital, el sistema solicitante, especificará que atributos de la identidad requiere y el usuario deberá confirmar dicha solicitud, siendo consciente de que información se está solicitando sobre su identidad.

Aunque aún es poco popular, OpenID está ganando fuerza debido al anuncio de algunos sitios importantes en la adopción de este sistema.

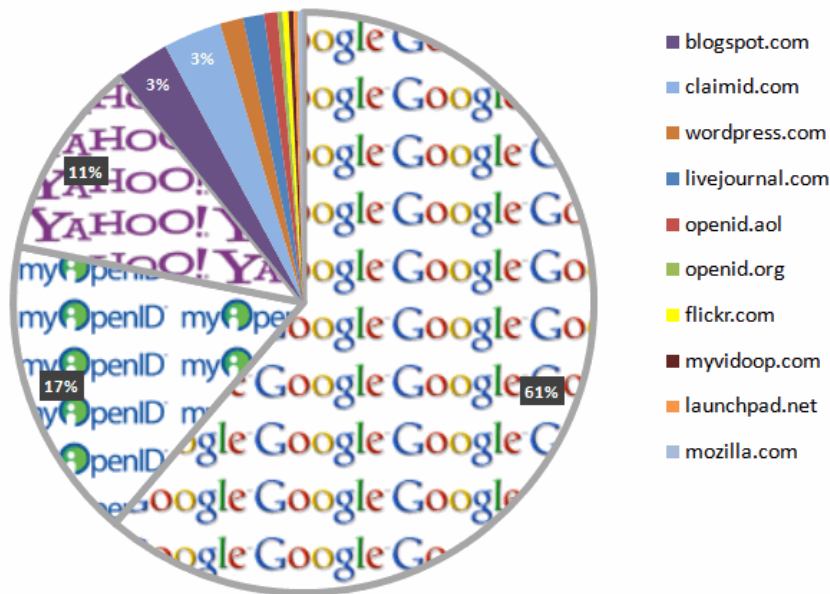


Figura 3: Gráfico de muestra con la utilización de cuentas OpenID

El anterior gráfico fue construido por uno de los administradores del popular foro sobre tecnologías de Software StackOverflow<sup>4</sup>. Muestra el porcentaje de uso de proveedores de identidad OpenID, por los usuarios que acceden al sitio.

## 2.4.Comunicaciones máquina a máquina<sup>5</sup>

Las comunicaciones máquina a máquina o M2M es un concepto genérico referido al intercambio de información entre dispositivos remotos. Aunque acostumbramos a identificar las comunicaciones como intercambios de información entre humanos como elementos finales del proceso, es conveniente hacer esta diferenciación y tratar las M2M como un campo de estudio con sus propias particularidades.

<sup>4</sup> <http://stackoverflow.com>

<sup>5</sup> <http://www.wikipedia.org>

Ejemplos de sistemas M2M pueden ser, gestión de flotas, alarmas domésticas, TPVs, contadores electrónicos, pantallas informativas, máquinas de Vending, tele mantenimientos, estaciones meteorológicas, y por supuesto servicios o protocolos en entornos Web. El marco de las M2M define los siguientes elementos.

- Dispositivo cliente M2M, como dispositivo o máquina diseñada para comunicar con el servidor y generalmente con capacidad de proceso para ejecutar la aplicación de negocio.
- Servidor M2M, máquina encargada de gestionar el envío y la recepción de la información enviada por los dispositivos, habitualmente integrado con los sistemas de gestión corporativos como ERPs.
- Red de comunicaciones, como el medio físico utilizado para el transporte de la información ya sea por cable o de forma inalámbrica.

Las comunicaciones M2M existen de alguna forma desde la aparición de las redes para los automatismos de carácter industrial, principalmente para aplicaciones SCADA, que datan incluso antes de la aparición de las comunicaciones móviles. En 1995 empieza a surgir esta red móvil, en el momento en que Siemens dedica un departamento para el desarrollo de un módulo de transmisión de datos sobre la red GSM, permitiendo a las máquinas acceder a esta red para el envío de información. Desde 1998 se desató un espectacular desarrollo sobre las M2M con importantes empresas de todo el mundo involucradas en el diseño de dispositivos dedicados a transferir información entre máquinas y el lanzamiento de satélites como plataformas de retransmisión de datos.

Actualmente las comunicaciones M2M y las redes inalámbricas culminan con la Nube, que sirve como plataforma para proveer soluciones M2M simplificando las redes de máquinas y permitiendo a los usuarios gestionarlas remotamente. A esto se le conoce como Platform as a Service, siendo una categoría de servicios de computación en la nube en la que el usuario controla el Software, su despliegue y la configuración, mientras que el proveedor pone a disposición la red, el Hardware y los servicios.

Se calcula que actualmente existen unas 187 millones de conexiones M2M en todo el mundo y según los estudios realizados, las aplicaciones M2M seguirán mostrando un importante crecimiento con unos ingresos estimados de 198 millones de dólares en 2018. En España la Comisión del Mercado de las Telecomunicaciones contabilizó 2,65 millones de líneas a mediados de 2012.

Existen empresas dedicadas a la investigación, desarrollo e innovación de sistemas M2M como Fraunhofer FOKUS y su equipo Fraunhofer M2M group. En su proyecto OpenMTC<sup>6</sup> advierten inconvenientes en las soluciones M2M actuales, como son las arquitecturas monolíticas y la falta de interoperabilidad, lo que impedirá a las redes M2M de gran escala

---

<sup>6</sup> <http://www.open-mtc.org>

desarrollarse correctamente. Proponen un esquema basado en la Plataforma OpenMTC destinado a grandes redes M2M como ciudades inteligentes o salud electrónica, en la que se abstraen las capas de comunicación mediante dispositivos OpenMTC y ofrecen un único interfaz a la capa de aplicación.

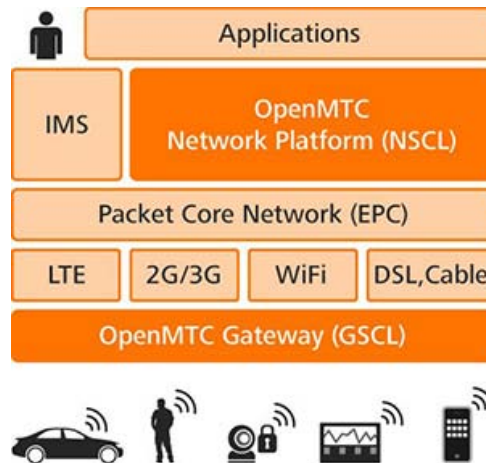


Figura 4: Diagrama de capas OpenMTC

# Capítulo 3

---

## 3. Descripción del sistema

A continuación se describen los aspectos funcionales y de arquitectura del sistema propuesto, es decir, como se estructuran los diferentes componentes que forman el sistema y que funciones cumplen dentro de él.

### 3.1.Arquitectura del sistema

El escenario se define como una entidad cliente que desea acceder a un recurso de una entidad servidora. Para ello la entidad servidora le especifica al cliente un servidor verificador de identidad a la que dirigirse junto con un reto. La entidad verificadora remite la respuesta al servidor del recurso, con la cual permitir o denegar el acceso al mismo.

#### 1. *Petición de acceso*

La entidad cliente desea acceder al recurso proporcionado por la entidad servidora. Para ello el cliente se dirige a la URL proporcionada por el servidor, la cual indica los datos del servicio de autenticación involucrado así como los datos de la transacción a realizar.

#### 2. *Petición de desafío*

Una vez el cliente es dirigido a la URL del servicio de autenticación, este se autentica en el sitio Web introduciendo su usuario y su contraseña. Se indicara al cliente si confía en la entidad servidora remitente y la conformidad para proporcionar a esta los datos solicitados de la cuenta.

#### 3. *Desafío resuelto*

Tras la autenticación del usuario y aceptada la conformidad, el cliente es dirigido al sitio Web del servidor, en el que se procesará la respuesta en base a los datos de la cuenta de usuario del cliente.

#### 4. *Respuesta de acceso*

El servidor procesa la respuesta, verificando los datos de la cuenta del cliente y los datos de la transacción proporcionados en la petición de acceso. Si todo es correcto, el servidor proporciona al cliente una respuesta positiva y muestra en local el mismo resultado.

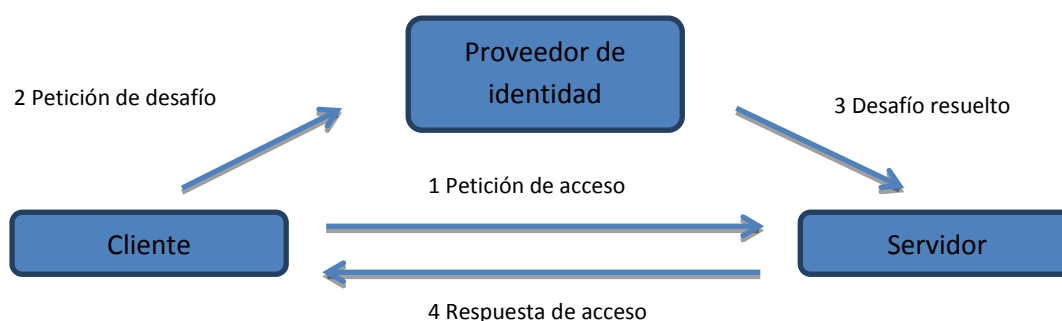


Figura 5: Esquema funcional del escenario propuesto

A continuación se definen las tecnologías involucradas en todos los procesos para el diseño de un posible escenario.

### 1. Cliente

El cliente es un dispositivo móvil, cargado con una aplicación de lectura de códigos QR de las muchas que se pueden encontrar para las distintas plataformas, junto a un navegador Web. Una de las virtudes del sistema planteado es que la tecnología o la plataforma del dispositivo no es determinante para el funcionamiento del sistema, lo que permite aprovechar el parque actual de Smartphones de las principales plataformas del mercado, como son Android e iOS.

### 2. Servidor

Equipo genérico. Carga una aplicación desarrollada en cualquier tecnología o lenguaje de alto nivel y orientado a objetos, que muestra en pantalla un código QR con la URL del proveedor de identidad y el reto, renovado con cada operación. Carga un servidor Web para la atención de las peticiones del servidor de identidad y tomar las decisiones sobre el permiso de acceso al recurso.

### 3. Proveedor de identidad

Servicio público de autenticación federada. Para ello será necesario abrir una cuenta de usuario, la cual se utilizará en todas las pruebas para la implementación del sistema.

Las credenciales de acceso serán las siguientes.

**Usuario:** <http://aingeru84.{provider}/>

**Contraseña:** Aingeru84

A continuación se muestra el esquema completo de la arquitectura del sistema. En él se especifican los pasos descritos y numerados por orden de ejecución, así como las entidades involucradas.

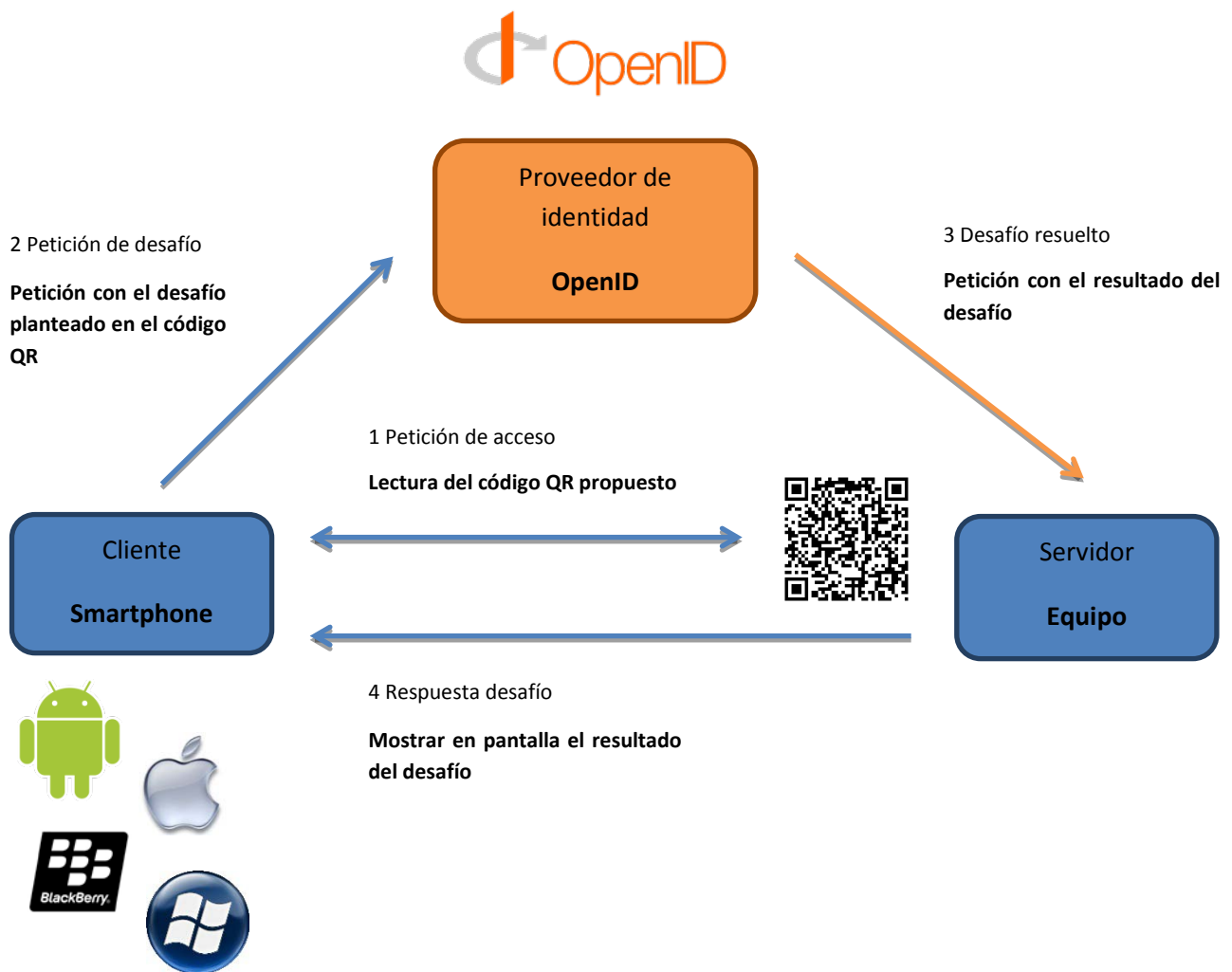


Figura 6: Esquema técnico del escenario propuesto

### 3.2.Descripción funcional

Para llevar a cabo el escenario planteado, es necesario implementar la aplicación que ejecutará la entidad servidora. A continuación se enumeran las tareas fundamentales que se deben realizar.

#### 1. Definición de la petición de desafío

El primer paso consiste en definir la petición que se realizará al proveedor de identidad. Esta petición se codificará en el código QR para el cliente, y marcará los parámetros que se



deben especificar. Esto nos permitirá conocer de antemano los parámetros necesarios y definir como obtenerlos.

Esta tarea es puramente documental, pero asienta las bases para la implementación de la aplicación, puesto que define que parámetros será necesario utilizar y por lo tanto de donde obtenerlos. Más adelante, una vez con todos los parámetros a nuestra disposición, podremos componer la petición de una forma sencilla.

El siguiente esquema muestra el diagrama de peticiones y respuestas de ejemplo llevadas a cabo por una entidad usuario, una entidad servidora y un proveedor de identidad. Componiendo un esquema de comunicaciones muy similar al del escenario planteado.

## *2. Carga de los parámetros de inicialización*

Una vez conocidos los parámetros necesarios para la composición de la petición y ciertos aspectos del funcionamiento de la aplicación, será necesario obtenerlos y ponerlos a disposición de la aplicación para su uso.

Una fase común en la mayoría de los proyectos Software consiste en la inicialización de la aplicación, es decir, cargar todos aquellos parámetros ya definidos y estructuras de datos que serán necesarios para el funcionamiento de la aplicación, así como las comprobaciones necesarias para verificar su correcto funcionamiento. La forma de obtener dichos parámetros puede ser muy diferente como se verá más adelante, pero deberán estar a disposición de la aplicación para realizar las tareas de implementación necesarias.

## *3. Generación de un código QR con el desafío propuesto*

Con la petición conformada y los parámetros disponibles, será necesario componer un código QR en base a dicha petición y mostrarlo en la interfaz de la aplicación para ponerlo a disposición de los dispositivos móviles y las aplicaciones de lectura de códigos QR.

La generación de códigos QR no es una capacidad estándar de los lenguajes de programación de alto nivel. Esto nos obliga a implementar dicha funcionalidad en base a la documentación sobre la tecnología QR, o si no queremos reinventar la rueda, acudir a componentes de terceros que cumplan dicha función.

## *4. Atención y respuesta de las peticiones de desafío resuelto*

Tras la autenticación del usuario del dispositivo móvil en el proveedor de identidad, este será redirigido a la máquina servidora, con lo que será necesario atender la petición de desafío resuelto y procesar la respuesta que se mostrará en el dispositivo y en la interfaz de la aplicación.

Podemos pensar que atender peticiones es una capacidad fundamental de servidores, pero dar la capacidad de atender dichas peticiones a una aplicación cualquiera no es especialmente complejo, solo requiere de la utilización de ciertos elementos del API del lenguaje que utilicemos y el manejo de la petición a bajo nivel.

### 3.3.Casos de uso

A continuación definiremos los casos de uso del sistema. Los casos de uso como descripción de las actividades que son necesarias para llevar a cabo un proceso, nos permitirán comprender los distintos estados definidos y los comportamientos que debe presentar nuestro sistema en cada momento.

#### 1. Caso

- Cliente realiza la lectura del código QR
- Servidor envía petición de ID de usuario
- Cliente cancela el proceso

#### 2. Caso

- Cliente realiza la lectura del código QR
- Servidor envía petición de ID de usuario
- Cliente introduce el identificador de usuario
- Proveedor de identidad envía petición de autenticación
- Cliente cancela el proceso

#### 3. Caso

- Cliente realiza la lectura del código QR
- Servidor envía petición de ID de usuario
- Cliente introduce el identificador de usuario
- Proveedor de identidad envía petición de autenticación
- Cliente introduce contraseña de usuario
- Proveedor de identidad envía petición de aceptación de condiciones
- Cliente cancela el proceso

#### 4. Caso

- Cliente realiza la lectura del código QR
- Servidor envía petición de ID de usuario
- Cliente introduce el identificador de usuario
- Proveedor de identidad envía petición de autenticación
- Cliente introduce contraseña de usuario
- Proveedor de identidad envía petición de aceptación de condiciones
- Cliente acepta condiciones del servicio

##### a. Caso

- Servidor evalúa código de operación incorrecta

**b. Caso**

- Servidor evalúa código de operación correcta
- Servidor calcula nuevo código de operación

**1.Caso**

- Servidor evalúa datos de usuario incorrectos

**2.Caso**

- Servidor evalúa datos de usuario correctos

Quedan definidos por tanto 6 casos de uso, que describen todos los posibles estados que puede presentar el sistema. Más adelante los casos de uso definidos nos servirán para especificar las pruebas necesarias y evaluar el correcto comportamiento de nuestro sistema.

# Capítulo 4

## 4. Implementación del sistema

En este capítulo se describe a nivel técnico las tecnologías y herramientas utilizadas así como el desarrollo de las tareas para la implementación del sistema propuesto. Todos los retos planteados por el escenario quedarán resueltos con su correspondiente justificación y gran nivel de detalle.

- El protocolo designado para realizar el proceso de autenticación federada será OpenID. En concreto el proveedor utilizado para el escenario será MyOpenID.
- La aplicación desplegada en el servidor será implementada utilizando tecnologías .Net, con el entorno de desarrollo Visual Studio 2010 y Team System como controlador de proyecto.

### 4.1.Petición OpenID

OpenID como protocolo de autenticación federada, define una serie de parámetros que se encuentran cuidadosamente descritos en las especificaciones publicadas por OpenID.net.

Tras analizar las especificaciones, a continuación se propone un patrón de petición con los parámetros requeridos para la implementación de nuestro sistema. Se incluye también la descripción de los parámetros utilizados.

```
https://www.myopenid.com/server
openid.claimed_id=http://Aingeru84.myopenid.com&
openid.identity= http://Aingeru84.myopenid.com &
openid.mode=checkid_setup&
openid.ns=http://specs.openid.net/auth/2.0&
openid.ns.sreg=http://openid.net/extensions/sreg/1.1&
openid.return_to=http://www.vending.com&
openid.sreg.required=nickname,email,fullname,postcode,country,language,dob&
openid.sreg.policy_url=http://www.clearbits.net/about/privacy
```

- **openid.claimed\_id** (opcional)  
Identificador de usuario solicitado. Este parámetro establece la cuenta o identidad que se requiere autenticar.
- **openid.identity** (opcional)  
Identidad que solicita la autenticación. La especificación indica que **openid.identity** y **openid.claimed\_id** deben estar ambos presentes u omitidos.
- **openid.mode**  
Especifica si el usuario quiere interactuar con el servidor directamente.

- **openid.ns**  
Indica la versión de la especificación OpenID a utilizar.
- **openid.ns.sreg**  
Valor fijo.
- **openid.return\_to** (opcional)  
Especifica al proveedor de identidad la URL a la que redirigir al cliente tras la autenticación. Si no se especifica, no es necesaria redirección.
- **openid.sreg.required** (opcional)  
Indica los datos de la cuenta de usuario que se solicitan.
- **openid.sreg.policy\_url** (opcional)  
URL indicada al usuario en la que se especifican las condiciones del servicio.

Una vez definida una petición de ejemplo y descritos los parámetros utilizados, es necesario parametrizar la petición para que sea independiente de valores como el identificador de usuario solicitado y la URL a la que es redirigido el cliente.

Un aspecto importante del proceso señalado en el escenario, es la identificación de la transacción realizada. Cuando la entidad servidora muestra el código QR con la petición a realizar, debe incluir un identificador de operación que se renovará con cada proceso. De esta forma la transacción queda correctamente identificada y permitirá a la máquina dar respuesta positiva solo a aquellos desafíos resueltos que incluyan el identificador de operación en curso, solo conocido por la máquina. Por tanto es necesario parametrizar la URL a la que se redirige al usuario con un prefijo que incluya la dirección IP en la que alcanzar la máquina y un código de operación como parámetro de dicha URL.

```
https://www.myopenid.com/server
openid.claimed_id=$user$&
openid.identity=$user$&
openid.mode=checkid_setup&
openid.ns=http://specs.openid.net/auth/2.0&
openid.ns.sreg=http://openid.net/extensions/sreg/1.1&
openid.return_to=$externalPrefix$/?operation_id=$operation$&
openid.sreg.required=nickname,email,fullname,postcode,country,language,dob&
openid.sreg.policy_url=http://www.clearbits.net/about/privacy
```

- **user**  
Identificador de usuario solicitado al proveedor de identidad. La máquina no puede conocer por sí sola el identificador de usuario que va a realizar la operación.
- **externalPrefix**

URL con la dirección IP accesible por el dispositivo cliente. La máquina debería disponer de la dirección IP desde la que puede ser accedida.

- **operation**

Identificador de la operación en curso. La máquina conoce en todo momento el identificador de operación que puede procesar.

El siguiente paso es realizar la carga de los parámetros sobre la petición OpenID. Los tres parámetros indicados tienen una naturaleza muy diferente y plantean problemas distintos que veremos a continuación.

El parámetro **user** es algo complicado. La máquina debe presentar al usuario un código QR con la petición OpenID que debe realizar, sin embargo no dispone del identificador de usuario en ningún momento. La única opción lógica es que el usuario introduzca su identificador, por lo tanto es necesario introducir un paso intermedio, en el que la máquina presente al usuario una sencilla página Web con código JavaScript, de forma que se le permita introducir en una caja de texto su identificador de usuario y al pulsar en Aceptar, se envíe la petición con el parámetro **user** especificado por el usuario. De esta forma conseguimos un efecto de Login en la máquina servidora, en la que adicionalmente se podría incluir información adicional como condiciones generales de venta o enriquecer la página con todo tipo de detalles.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Registro</title>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    <style type="text/css">
      #user {
        width: 292px;
      }
      .style1
      {
        height: 53px;
      }
    </style>
  </head>
  <body>
    <script type="text/javascript">
      function RedirectToUrl() {

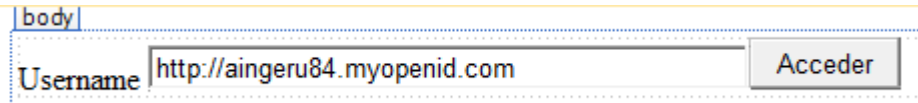
        var urlCode = "https://www.myopenid.com/server?" +
          "openid.claimed_id=$user$" +
          "openid.identity=$user$" +
          "openid.mode=checkid_setup" +
          "openid.ns=http://specs.openid.net/auth/2.0" +
          "openid.ns.sreg=http://openid.net/extensions/sreg/1.1" +
          "openid.return_to=$externalPrefix$/?operation_id=$operation$" +
          "openid.sreg.required=nickname,email,fullname,postcode,country,language,dob" +
          "openid.sreg.policy_url=http://www.clearbits.net/about/privacy" +
          "no_ssl=true";

        urlCode =urlCode.replace("$user$",document.getElementById("user").value)
        urlCode =urlCode.replace("$user$",document.getElementById("user").value)
```

```
        window.location.replace(urlCode);
    }
</script>

<form method="get" >
    Username <input type="text" id="user" name="user"
value="http://aingeru84.myopenid.com"/>
    <button type="button" onclick="RedirectUrl()">Acceder</button>
</form>

</body>
</html>
```



La carga del parámetro **externalPrefix** consiste en averiguar la dirección o direcciones IP en las cuales la máquina puede ser accedida. Dicho esto caben dos opciones, acceder a través de la dirección IP en la red local o la dirección pública de la red local desde la cual acceder por internet. En este último caso, es importante tener en cuenta que será necesario configurar en el dispositivo enrutador o router de la red local, una redirección del puerto utilizado o NAT (en nuestro caso en el puerto por defecto del protocolo HTTP que es el 80) hacia la dirección IP de la máquina en la red local.

Para averiguar la dirección IP de la red local, basta con que la aplicación busque la configuración establecida entre los adaptadores de red locales, lo cual es muy sencillo. Para averiguar la dirección IP externa de la red, nos dirigiremos a servicios de consulta de IP externa, como pueden ser WhatIsMyIP o DynDNS. En nuestro caso nos dirigimos a DynDNS que dispone de dicho servicio gratuito, puesto que actualmente para WIMI es necesario crear una cuenta de usuario. Técnicamente es sencillo, basta con lanzar una petición HTTP al servidor con la URL especificada, y procesar la respuesta a través de una expresión regular para extraer el dato que buscamos. A continuación se muestra el código .Net para alcanzar dichos objetivos.

```
Private Shared Function GetLocalIP() As String

    ' Search for specific network adapter
    For Each address As System.Net.IPAddress In
System.Net.Dns.GetHostEntry(System.Net.Dns.GetHostName()).AddressList
        If address.AddressFamily = Net.Sockets.AddressFamily.InterNetwork Then
Return address.ToString
        Next

    ' Not found
    Return Nothing

End Function
```

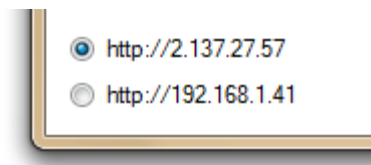
```
Public Shared Function GetExternalIPDynDNS() As String

    Dim textResponse As String

    ' Gets data bytes for the http response to the specified url
```

```
Using response As WebResponse =  
HttpRequest.Create("http://checkip.dyndns.com/").GetResponse()  
Using reader As New BinaryReader(response.GetResponseStream())  
textResponse =  
System.Text.Encoding.ASCII.GetString(reader.ReadBytes(1024))  
End Using  
End Using  
  
' Extract address from server response  
textResponse = System.Text.RegularExpressions.Regex.Match(textResponse,  
"<body>Current IP Address:(?<address>.*</body>").Groups("address").Value  
Return textResponse.Trim  
  
End Function
```

Para realizar las pruebas oportunas, la aplicación permite establecer el prefijo que se desea utilizar seleccionando uno u otro a través del interfaz de la aplicación.



El parámetro **operation** es trivial, la aplicación carga un identificador numérico inicial de forma aleatoria, que es renovado cada vez que se procesa un desafío resuelto, recalculando así la nueva petición.

## 4.2. Componente de codificación QR

Como se ha mencionado anteriormente, los lenguajes de alto nivel no disponen de la capacidad de generar códigos QR. Para ello hemos acudido al API Chart de Google, muy bien documentado y ampliamente utilizado. El único aspecto relevante en cuanto a este servicio, radica en que es un servicio Web, con lo que la generación del código QR se computa en los servidores de Google, y no en la máquina local como cabría esperar. No obstante el escenario plantea la necesidad obligatoria de estar conectado a internet para poder enviar la petición a los servicios OpenID y de la misma forma recibir los desafíos resueltos, con lo que no supone ningún inconveniente adicional.

La implementación de este requisito ha tomado la forma de componente, o mejor dicho, hablando en términos de lenguaje .Net, de control al que se ha denominado QRCodeBox. Un control .Net es una clase con representación visual, la cual se puede reutilizar y editar visualmente con la herramienta de WinForms Designer de Visual Studio. De esta forma el control de codificación QR es independiente del propósito y reutilizable desde la librería en la que se encuentra por otros proyectos.

La clase dispone de una función estática que realiza la tarea de enviar la petición de conversión al servicio de Google y transformar los datos binarios recibidos en un objeto de tipo



Image de .Net. La URL que conforma la petición queda extraída en una constante parametrizada para su uso dentro de la función.

```
Private Const QRCODE_REQUEST As String = "http://chart.apis.google.com/chart?cht=qr&chs={0}x{1}&chld=L|0&chl={2}"

''' <summary>
''' Gets de object image for the specified data on a QR code.
''' </summary>
''' <param name="data">Unencoded URL data for the QR code generation.</param>
''' <param name="height">Desired height for the QR code</param>
''' <param name="width">Desired width for the QR code</param>
''' <returns>Image object of the generated QR code.</returns>
Public Shared Function GenerateQR(ByVal data As String, ByVal height As Integer, ByVal width As Integer) As Image
    Try
        Dim bytes As Byte()

        ' Access Google Charts API to get the QR Code chart bytes
        Using response As WebResponse = HttpWebRequest.Create(String.Format(QRCODE_REQUEST, height, width, data)).GetResponse()
            Using reader As New BinaryReader(response.GetResponseStream())
                bytes = reader.ReadBytes(response.ContentLength)
            End Using
        End Using

        ' Get the resulting image from the response bytes
        Return Image.FromStream(New MemoryStream(bytes))

    Catch ex As Exception
        Throw New Exception("Error generating QR Code", ex)
    End Try
End Function
```

Durante las pruebas encontramos correcta la función planteada que integra con el servicio Chart de Google. Sin embargo aparecen dificultades a la hora de codificar datos extensos, produciendo que la densidad de puntos del código QR no sea adecuada para mostrar en la interfaz de la aplicación, no ser legible por dispositivos móviles y en casos extremos que el servicio nos indique un código de error indicativo de mensaje demasiado largo. Para ello acudimos al servicio de reducción de URL de Google, dicho servicio recibe el nombre de Goo.gl.

Según se indica en la documentación, dicho servicio se puede utilizar de forma libre con ciertas limitaciones o con limitaciones ampliadas registrando la aplicación a través de Google y recibiendo una clave para la llamada del servicio. En nuestro caso con la poca utilización que se le dará al servicio, no consideramos necesario realizar ningún registro.

La llamada al servicio es bien sencilla según las especificaciones. Transportar dichas especificaciones a código .Net no es difícil, solo es necesario componer la petición POST al servicio y extraer el dato relevante de la respuesta mediante una expresión regular.

```
POST https://www.googleapis.com/urlshortener/v1/url
Content-Type: application/json

{"longUrl": "http://www.google.com/"}
```

```
''' <summary>
''' Access the Googl service for URL shortening
''' </summary>
''' <param name="longUrl">Long URL for being shortened</param>
''' <returns>Shortened URL</returns>
Private Shared Function GetShortUrl(ByVal longUrl As String, Optional ByVal key As Str
ing = Nothing) As String
    Try
        ' Validate long URL to avoid invalid calls
        Dim uri As New Uri(longUrl)

        ' Set header request
        Dim request As HttpWebRequest
        If Not String.IsNullOrEmpty(key) Then
            request = HttpWebRequest.Create(String.Format(SHORTEN_REQUEST_KEY, key))
        Else
            request = HttpWebRequest.Create(SHORTEN_REQUEST)
        End If
        request.Method = "POST"
        request.ContentType = "application/json"
        'request.ContentLength = data.Length

        Dim data() As Byte = System.Text.Encoding.ASCII.GetBytes(String.Format("{0}" & lo
ngUrl, "{0}"))

        ' Set body request
        Using writer As Stream = request.GetRequestStream
            writer.Write(data, 0, data.Length)
        End Using

        ' Get response from request
        Using response As WebResponse = request.GetResponse()
            Using reader As New StreamReader(response.GetResponseStream())

                ' Get matching parameter on the response for the short url
                With System.Text.RegularExpressions.Regex.Match(reader.ReadToEnd(), ""
"id": "(?<shortUrl>.{0,})")
                    If .Success Then
                        Return .Groups("shortUrl").Value()
                    Else
                        Return longUrl
                    End If
                End With
            End Using
        End Using

    Catch ex As Exception
        Return longUrl
    End Try
End Function
```

A continuación, se integran las implementaciones realizadas en la clase control de .Net, estableciendo las características del control para que se comporte como tal en vista de diseño sobre Visual Studio. El funcionamiento que debe ofrecer es sencillamente mostrar en la interfaz una imagen del código QR que codifica los datos especificados en la programación, con lo que es trata fundamentalmente de un control PictureBox .Net con un funcionamiento interno específico. Dicho esto, basta con hacer que la clase QRCodeBox herede del control nativo PictureBox, y ya dispondremos del funcionamiento en forma de control. El paso

siguiente es agregar las funcionalidades de generación de código QR y acortamiento a la clase e implementar las propiedades y métodos necesarios para manejar de forma apropiada el control desde la aplicación.

Cabe señalar que se han añadido dos propiedades al control. La propiedad Data contiene los datos codificados en el último código QR generado. ShortUrl especifica si se debe acortar o no la URL especificada para codificar. Hay que tener en cuenta que aunque se puede codificar cualquier tipo de información sobre un código QR, el servicio de reducción de longitud de URLs solo permite trabajar con información o datos de tipo URI.

```
''' <summary>
''' Main method for loading QR Code of the specified data.
''' </summary>
''' <param name="data">Data for being codified on th QR Code</param>
''' <returns>Data result of the QR Code</returns>
Public Function LoadQR(ByVal data As String) As String

    ' Short Url if necessary
    If _shortUrl Then _data = GetShortUrl(data) Else _data = data

    ' Set QR Code as image
    Me.BackgroundImage = GenerateQR(Web.HttpUtility.UrlEncode(_data), Me.Height, Me.Width)

    Return _data

End Function
```

El control de generación de códigos QR está completo y a disposición de la aplicación para hacer uso de él. Al compilar el proyecto de aplicación, dispondremos de una nueva opción de control QRCodeBox en la barra de herramientas, el cual podremos arrastrar cómodamente al formulario que deseemos utilizar como interfaz de la aplicación, y desde la edición del código fuente cargar los datos oportunos cuando sea necesario.

### 4.3.Atención de peticiones HTTP

La aplicación de la máquina servidora debe atender dos tipos de peticiones HTTP, por un lado las peticiones de Login, tal y como se ha comentado con anterioridad, y por otro las peticiones de redirección por el desafío OpenID resuelto. La tecnología .Net al igual que otras, ofrece un API muy extenso para el manejo de las comunicaciones en redes a distintos niveles de manejo, desde sockets TCP a bajo nivel hasta hilos de escucha sobre HTTP, todo ello bajo el espacio System.Net.

En nuestro caso nos hemos inclinado por el uso de hilos de escucha TCP System.Net.TCPListener, que facilitan mucho el control de la escucha de peticiones al manejar

internamente las gestión multihilo, permitiéndonos al mismo tiempo un control más plano y abierto a bajo nivel sobre los datos enviados y recibidos.

Para ello hemos realizado una implementación propia de la clase mencionada, encapsulando un objeto de dicho tipo y manejando de forma específica las peticiones y las respuestas esperadas. Al recibir respuestas, la clase notifica a través de eventos asíncronos los datos recibidos de forma que las clases contenedoras y preparadas para escuchar dichos eventos puedan operar en consecuencia. Adicionalmente se ha incluido un registro de las peticiones y respuestas producidas por las escucha sobre la salida estándar de la aplicación para facilitar la depuración de las mismas.

De esta forma el manejo del escuchador de peticiones es bien sencillo. Basta con instanciar un objeto de la clase, indicando la escucha de sus eventos producidos y arrancar la escucha.

```
Private WithEvents _listener As CustomListenerTcp

Private Sub Initialize()

    ' Start service
    _listener = New CustomListenerTcp(_internalPrefix, _listenPort)
    _listener.Start()

End Sub
```

Al arrancar la escucha de peticiones, la clase opera con un método que en esencia es bastante genérico, pero transforma las peticiones recibidas en eventos tipados, haciendo que el trabajo con ellas sea claro y sencillo.

```
Private Sub Listen()
    Try
        ' Enter the listening loop.
        While True
            With CType(_server.LocalEndPoint, IPEndPoint)
                Console.WriteLine(String.Format("{0}:{1}) Waiting for a connection...", .Address, .Port))
            End With

            ' Perform a blocking call to accept requests
            Using client As TcpClient = _server.AcceptTcpClient()
                With CType(client.Client.RemoteEndPoint, IPEndPoint)
                    Console.WriteLine(String.Format("{0}:{1}) Connected!" & vbCrLf, .Address, .Port))
                End With
                System.Threading.Thread.Sleep(1500)

                ' Get a stream object for reading and writing
                Using stream As NetworkStream = client.GetStream()

                    ' Buffer for reading data
                    Dim bytes(1024) As Byte
                    Dim data As String = String.Empty

                    While stream.DataAvailable

                        ' Loop to receive all the data sent by the client
                        Dim i As Int32 = stream.Read(bytes, 0, bytes.Length)
```

```
' Translate data bytes to a ASCII string
data += System.Text.Encoding.ASCII.GetString(bytes, 0, i)

End While

' Set response action
Console.WriteLine(data)
System.Threading.Thread.Sleep(1500)
ProcessRequest(client, data)

End Using

End Using

End While

Catch ex As SocketException
    Console.WriteLine("SocketException: {0}", ex)
End Try
End Sub

Private Sub ProcessRequest(ByVal client As TcpClient, ByVal data As String)
    Try
        If Not String.IsNullOrEmpty(data) Then
            If data.StartsWith("GET /?login") Then
                RaiseEvent LoginResponse(Me, New CustomResponseEventArgs(client, data))
            )

            ElseIf data.StartsWith("GET /?operation_id") Then
                RaiseEvent OpenIdResponse(Me, New CustomResponseEventArgs(client, data))
            ))

            Else
                Throw New Exception("Unknown request")
            End If
        End If
    End If

    Catch ex As Exception
        Console.WriteLine("ProcessingException: {0}", ex)
    End Try
End Sub
```

La gestión de los eventos producidos es trivial en estos casos, solo es necesario la declaración de la función delegada del evento y la asignación de su manejador de evento. Dentro de la función delegada, siguiendo el estándar .Net los datos recibidos se encapsulan en un objeto de implementación propia, que expone los datos recibidos y especifica las acciones que se pueden realizar, entre las cuales esta producir una determinada respuesta.

```
Private Sub _listener_OpenIdResponse(ByVal sender As Object, ByVal e As CustomResponse
EventArgs) Handles _listener.OpenIdResponse

    ' Send back operation result
    e.SetOperationResponse(e.OperationID = _operationId)

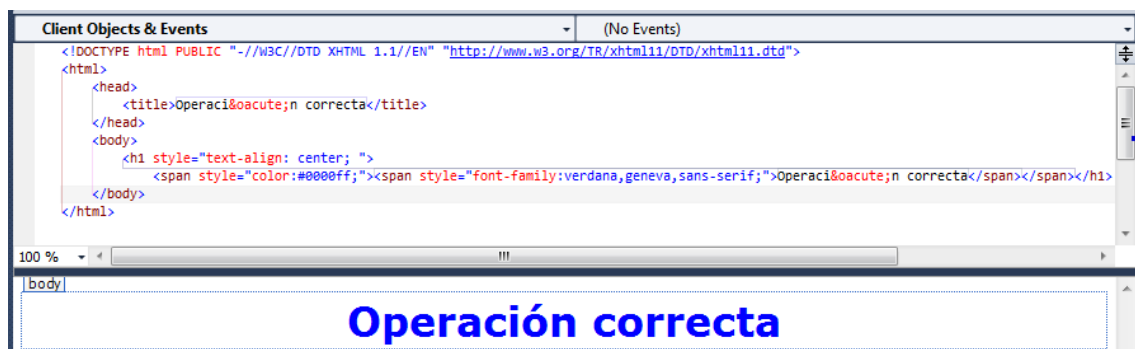
End Sub

Private Sub _listener_LoginResponse(ByVal sender As Object, ByVal e As CustomResponseE
ventArgs) Handles _listener.LoginResponse

    ' Send back login form
    If lblPrefijoExterno.Checked Then
        e.SetLoginResponse(_operationId, _externalPrefix)
    Else
        e.SetLoginResponse(_operationId, _internalPrefix)
```

`End If``End Sub`

Ya disponemos de la clase de escucha de peticiones HTTP. Su utilización como se ha descrito es sencilla y solo falta indicar las respuestas enviadas. Visual Studio dispone de un potente editor de documentos HTML mediante el cual se han definido las respuestas, siendo susceptibles de mejorarse y enriquecerse todo lo que se requiera. Una vez definidas, se incluyen como recursos de la aplicación quedando embebidos y disponibles para su utilización inmediata a través del código fuente autogenerado por Visual Studio en forma de propiedades globales.



```
Public Sub SetOperationResponse(ByVal valid As Boolean)

    Dim response As String

    ' Build response
    If valid Then
        response = String.Format("HTTP/1.1 200 OK{0}Content-
Type: text/html; charset=utf-8{0}Content-
Length: length{0}{0}{1}", vbCrLf, My.Resources.OpSuccess)
    Else
        response = String.Format("HTTP/1.1 200 OK{0}Content-
Type: text/html; charset=utf-8{0}Content-
Length: length{0}{0}{1}", vbCrLf, My.Resources.OpFail)
    End If
    Console.WriteLine(response)

    ' Write response to the stream
    Dim bytes As Byte() = System.Text.Encoding.ASCII.GetBytes(response)
    Using writer As IO.Stream = _response.GetStream
        writer.Write(bytes, 0, bytes.Length)
    End Using

End Sub
```

#### 4.4.Integración de funcionalidades

Ahora es el momento de poner en común todas las funcionalidades implementadas, para cubrir el escenario propuesto. Para ello se ha desarrollado la aplicación QRServer, que se desplegaría sobre la máquina servidora. Según lo descrito en el análisis funcional, su misión es presentar un código QR al usuario del dispositivo móvil, de forma que al realizar la lectura sea redirigido a un proveedor de identidad y tras realizar la autenticación, será redirigido de nuevo a la máquina. La aplicación recogerá la petición de redirección y evaluará el resultado de la operación, que se mostrará tanto en el dispositivo móvil como en la interfaz de la aplicación.

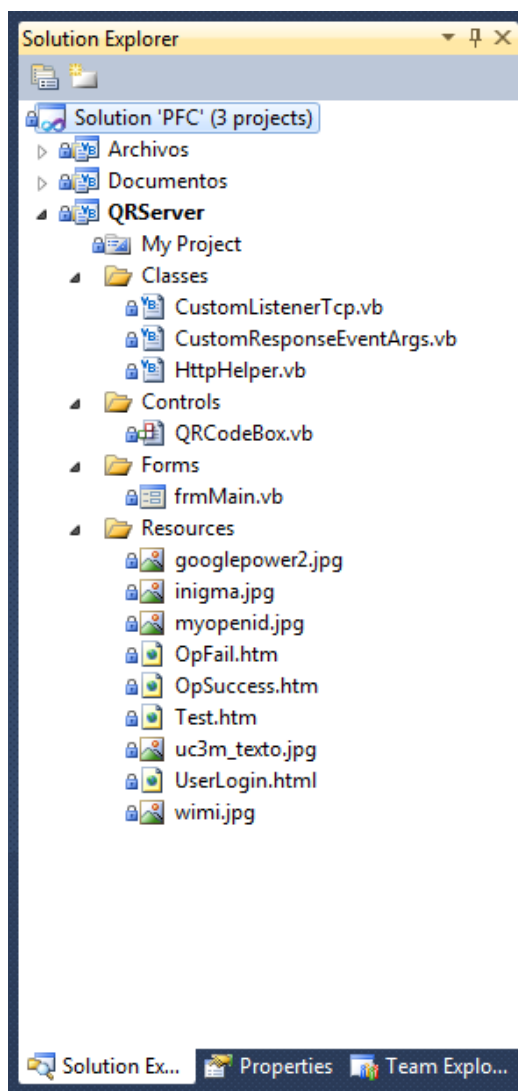


Figura 7: Estructura del proyecto en el entorno de desarrollo

En la figura se puede apreciar la sencilla estructuración de los elementos que conforman el proyecto Software. Entre ellos podemos encontrar el control de códigos QR bajo el fichero QRCodeBox.vb y los ficheros que conforman el manejador de peticiones Http, bajos los ficheros CustomListenerTcp.vb y CustomResponseEventArgs.vb.

La aplicación solo necesita de un formulario .Net como interfaz. En él se mostrará el código QR y los datos del usuario que está realizando la operación en el momento de procesar la

resolución del desafío, así como el resultado de dicha operación. Como se ha mencionado anteriormente, disponemos de un selector para especificar el prefijo de red interno o externo, lo cual facilita las pruebas realizadas. Adicionalmente, se han incluido unos enlaces a modo de sponsor para mostrar los servicios o entidades con los que hemos integrado.

El código fuente para implementar lo mencionado anteriormente es sencillo, puesto que los controles desarrollados han sido diseñados pensando en la simplicidad de su uso. De hecho basándonos en los códigos fuente de muestra indicados en el punto anterior, su uso es bastante similar. A continuación se detalla la definición de la clase formulario principal, así como las funciones más relevantes, como son la inicialización y el manejo de las peticiones Http.

```
''' <summary>
''' Clean user interface
''' Load network prefixes
''' Start Tcp service
''' Load QR Code
''' </summary>
Private Sub Initialize()

    CheckForIllegalCrossThreadCalls = False

    ' Clean UI
    lblTransaccion.Text = String.Empty
    lblFullName.Text = String.Empty
    lblMail.Text = String.Empty
    lblDoB.Text = String.Empty
    lblAge.Text = String.Empty

    Try
        ' Load external prefix
        _externalPrefix = NetHelper.GetExternalIPDynDNS()
        lblPrefijoExterno.Text = String.Format("http://{0}:{1}", _lis
tenPort)
    Catch ex As Exception
        lblPrefijoExterno.Text = "Error al configurar IP externa"
    End Try

    Try
        ' Load internal prefix
        _internalPrefix = NetHelper.GetLocalIP()
        lblPrefijoInterno.Text = String.Format("http://{0}:{1}", _lis
tenPort)
    Catch ex As Exception
        lblPrefijoInterno.Text = "Error al configurar IP interna"
    End Try

    ' Start service
    _listener = New CustomListenerTcp(_internalPrefix, _listenPort)
    _listener.Start()

    ' Set initial code
    LoadQRCode()

End Sub
```

```
Private Sub _listener_OpenIdResponse(ByVal sender As Object, ByVal e As CustomResponse
EventArgs) Handles _listener.OpenIdResponse
```



```
' Send back operation result
e.SetOperationResponse(e.OperationID = _operationId)

' Show operation result
If e.OperationID = _operationId Then
    lblTransaccion.Text = "Operación correcta"
    lblTransaccion.ForeColor = Color.Blue
    LoadQRCode()
Else
    lblTransaccion.Text = "Operación incorrecta"
    lblTransaccion.ForeColor = Color.Red
End If
lblTransaccion.Refresh()

' Set user account values
lblFullName.Text = e.OpenIFullName
lblMail.Text = e.OpenIEMail
lblDoB.Text = e.OpenIdDoB
lblAge.Text = IIf(Now.Year -
e.OpenIdDoB.Year >= 18, "Mayor de edad", "Menor de edad")
lblFullName.Refresh()
lblMail.Refresh()
lblDoB.Refresh()
lblAge.Refresh()

tmrClean.Start()

End Sub

Private Sub _listener_LoginResponse(ByVal sender As Object, ByVal e As CustomResponseE
ventArgs) Handles _listener.LoginResponse

    ' Send back login form
    If lblPrefijoExterno.Checked Then
        e.SetLoginResponse(_operationId, lblPrefijoExterno.Text)
    Else
        e.SetLoginResponse(_operationId, lblPrefijoInterno.Text)
    End If

End Sub
```

Como resultado final, a continuación se muestra el aspecto general de la aplicación en tiempo de ejecución, preparada para recibir peticiones de usuario y procesar respuestas de desafío resuelto.



Figura 8: Aspecto visual de la aplicación en tiempo de ejecución

# Capítulo 5

## 5. Pruebas

En este capítulo se definen las pruebas a las que se someterá el sistema para verificar su viabilidad y su correcto funcionamiento. Un buen plan de pruebas automatizadas permite evaluar de forma eficaz y segura el correcto funcionamiento de nuestro sistema, evitando realizar tareas de comprobación manuales. Dada la interoperabilidad del sistema con servicios externos será necesario definir un plan de pruebas de integración. Se establecerán unas sencillas pruebas de sistema para demostrar el funcionamiento multiplataforma desde el lado del cliente. Y por último, habiendo descrito los casos de uso, se plantearán los casos de prueba asociados.

### 5.1. Pruebas de integración

El sistema desarrolla su actividad utilizando servicios y funcionalidades externas, dependiendo fundamentalmente de ellas. Por ello resulta imprescindible diseñar un plan de pruebas de integración sobre dichas dependencias externas, para poder conocer en todo momento el estado de la relación entre nuestro sistema y los servicios consumidos.

#### 1. Servicio OpenID

El servicio OpenID proporcionado por MyOpenID es la piedra angular de nuestro sistema, por tanto es vital conocer el estado del servidor que nos proporciona el servicio. Para ello una buena forma de verificarlo, es enviando la petición parametrizada de nuestro sistema y analizar la respuesta en busca de los indicadores de respuesta correcta desde el servidor, puesto que si la petición es incorrecta o está mal formada, lo advertiremos en la respuesta recibida. La prueba diseñada para verificar este servicio consta de las siguientes validaciones.

- Verificar servidor operativo
  - Servidor en línea
  - Sitio web en línea
- Verificar respuesta a petición de cliente
  - Respuesta correcta
  - Parámetros especificados correctos

La implementación de esta prueba se basa en el método de prueba descrito a continuación.

```
<TestMethod()>
```

```
Public Sub OpenIDIntegration()  
    Try  
        Dim user As String = "http://aingeru84.myopenid.com"  
        Dim externalPrefix As String = "http://" & NetHelper.GetExternalIPDynDNS() & "  
        : " & 82  
        Dim operationId As Integer = New Random().Next  
        Dim textResponse As String  
        Dim request As String = "https://www.myopenid.com/server?" +  
            "openid.claimed_id=$user$&" +  
            "openid.identity=$user$&" +  
            "openid.mode=checkid_setup&" +  
            "openid.ns=http://specs.openid.net/auth/2.0&" +  
            "openid.ns.sreg=http://openid.net/extensions/sreg/1.1&" +  
            "openid.return_to=$externalPrefix$?operation_id=$operation$&" +  
            "openid.sreg.required=nickname,email,fullname,postcode,country,language,dob&" +  
            "openid.sreg.policy_url=http://www.clearbits.net/about/privacy&" +  
            "no_ssl=true"  
        request = request.Replace("$user$", user)  
        request = request.Replace("$externalPrefix$", externalPrefix)  
        request = request.Replace("$operation$", operationId)  
        ' Service request  
        Using response As WebResponse = HttpWebRequest.Create(request).GetResponse()  
            ' Check received response  
            Assert.IsTrue(DirectCast(response, System.Net.HttpWebResponse).StatusDescription = "OK")  
            Using reader As New StreamReader(response.GetResponseStream(), Encoding.GetEncoding(1252))  
                textResponse = reader.ReadToEnd  
            End Using  
        End Using  
        ' Check received response  
        Assert.IsTrue(textResponse.Contains("Sign In"))  
    Catch ex As Exception  
        Throw  
    End Try  
End Sub
```

## 2. Servicio GoogleChart

Google proporciona el servicio Google Chart, el cual nos permite generar los códigos QR de una forma muy sencilla enviando una petición a su servicio. Aunque es difícil que los servicios de Google no respondan, o incluso que las especificaciones varíen, sí que es posible que realicemos cambios en nuestra implementación que comprometan el funcionamiento, resultando que nuestro sistema no pueda llevar a cabo la presentación del código QR. Por ello es necesario diseñar una sencilla prueba que valide el servicio. Adicionalmente se diseña una prueba similar para verificar el funcionamiento del servicio de acortamiento de URLs.

- Verificar respuesta a petición Google Chart del servidor
  - Respuesta correcta
- Verificar respuesta a petición Goo.gl del servidor
  - Respuesta correcta

La implementación de esta prueba se basa en los métodos de prueba descritos a continuación.

```
<TestMethod()>
Public Sub QRCodeIntegration()
    Try
        Dim height As Integer = 20
        Dim width As Integer = height

        ' Service request
        Dim image As System.Drawing.Image = QRCodeBox.GenerateQR("test", height, width
    )

        ' Check received response
        Assert.IsTrue(image.Height = height AndAlso image.Width = width)

    Catch ex As Exception
        Throw
    End Try
End Sub

<TestMethod()>
Public Sub UrlShortenerIntegration()
    Try
        Dim longUrl As String = "https://www.myopenid.com/"

        ' Service request
        Dim shortUrl As String = QRCodeBox.GetShortUrl(Longurl)

        ' Check received response
        Assert.IsTrue(shortUrl.Length < longUrl.Length)

    Catch ex As Exception
        Throw
    End Try
End Sub
```

### 3. Servicio DynDNS

El servicio gratuito de descubrimiento de IP proporcionado por DynDNS es necesario para que nuestro sistema pueda parametrizar la dirección IP a la que se debe dirigir al cliente para procesar la respuesta. El principal riesgo que sufre este servicio es que al ser proporcionado por una empresa privada, dicho servicio se deshabilite o se convierta en un servicio de pago. De hecho, anteriormente esta funcionalidad se conseguía a través del servicio de WhtalsMylp, el cual dejó de ser de libre acceso, requiriendo una cuenta de usuario en el sitio Web. Afortunadamente este servicio no es vital, pudiendo sustituirse por una configuración en la máquina local como solución alternativa.

- Verificar respuesta a petición del servidor
  - Respuesta correcta

La implementación de esta prueba se basa en el método de prueba descrito a continuación.

```
<TestMethod()>
Public Sub ExternalIPIntegration()
    Try
        ' Service request
        Dim externalIp As String = QRServer2.NetHelper.GetExternalIPDynDNS()

        ' Check received response
        Assert.IsTrue(NetHelper.IsPublicIp(IPAddress.Parse(externalIp).ToString))

    Catch ex As Exception
        Throw
    End Try
End Sub
```

## 5.2.Pruebas de sistema

En este apartado contemplamos las pruebas realizadas sobre distintos tipos de hardware o plataforma. Esto nos servirá como un pequeño registro de los dispositivos o plataformas utilizadas para comprobar el funcionamiento de nuestro sistema. La principal motivación de este tipo de pruebas es demostrar que la definición funcional del lado del cliente, es en efecto, independiente del resto del sistema, pudiendo aprovecharse el gran abanico de dispositivos móviles y aplicaciones de lecturas de códigos QR que hay en el mercado.

La aplicación diseñada para la parte de servidor está desarrollada sobre tecnología .Net y por tanto se ejecuta bajo .Net Framework, lo cual aísla su funcionamiento del dispositivo sobre el que se monte. Esto excluye cualquier tipo de prueba de sistema que queramos describir sobre esta parte, ya que se asume que cualquier dispositivo, equipo o máquina, independientemente de cualquier factor, si dispone de una distribución de .Net Framework, es susceptible de montar la aplicación proporcionando el mismo funcionamiento.

En el lado de cliente disponemos de un gran número de dispositivos, plataformas, y aplicaciones sobre las que realizar pruebas, cuya permutación nos puede arrojar un exceso innecesario de resultados para cumplir nuestro objetivo. Tendremos en cuenta un pequeño número de dispositivos sobre los que realizar pruebas, aprovechando diferentes plataformas y de la misma forma un pequeño número de aplicaciones de lectura QR.

Aplicaciones de lectura de QR

- Inigma3Gvision
- BIDI



Dispositivos móviles

- iPhone 4S 16Gb
  - IOS
- Sony Xperia U

- Android
- Blackberry Curve 3G 9300
  - BlackBerry OS

### 5.3.Casos de prueba

Los casos de prueba definen la batería de pruebas a realizar, clasificadas y jerarquizadas para poner a prueba nuestro sistema y disponer de un cuadrante con las tareas a realizar y los resultados esperados. Normalmente los casos de prueba suelen atender a pruebas funcionales automatizadas, es decir, simulaciones automáticas sobre el uso de las aplicaciones, pero en nuestro caso es difícil de aplicar de esta manera debido a la naturaleza de nuestro sistema cliente – servidor.

Basándonos en los casos de uso planteados, podemos componer los casos de prueba asociados y desglosar las tareas y los resultados que debemos esperar. Identificaremos cada caso de prueba mediante una sencilla codificación para manejar de forma eficaz las entradas en la tabla que compondremos.

Caso de prueba	Caso de uso	Tareas	Resultado
P1	Caso 1	<ul style="list-style-type: none"><li>• Realizar la lectura del código QR con un dispositivo móvil</li></ul>	El sistema no debe realizar ninguna operación
P2	Caso 2	<ul style="list-style-type: none"><li>• Realizar la lectura del código QR con un dispositivo móvil</li><li>• Introducir identificador de usuario</li></ul>	El sistema no debe realizar ninguna operación
P3	Caso 3	<ul style="list-style-type: none"><li>• Realizar la lectura del código QR con un dispositivo móvil</li><li>• Introducir identificador de usuario</li><li>• Introducir contraseña de usuario</li></ul>	El sistema no debe realizar ninguna operación
P4	Caso 4 – a	<ul style="list-style-type: none"><li>• Realizar la lectura del código QR con un dispositivo móvil</li><li>• Introducir identificador de usuario</li></ul>	El sistema de mostrar los datos de la cuenta de usuario recibida y un mensaje de operación incorrecta.

		<ul style="list-style-type: none"> <li>• Introducir contraseña de usuario</li> <li>• Aceptar condiciones del servicio</li> <li>• Repetir la operación realizada</li> </ul>	
P5	Caso 4 – b – 1	<ul style="list-style-type: none"> <li>• Realizar la lectura del código QR con un dispositivo móvil</li> <li>• Introducir identificador de usuario sin mayoría de edad</li> <li>• Introducir contraseña de usuario</li> <li>• Aceptar condiciones del servicio</li> </ul>	El sistema de mostrar los datos de la cuenta de usuario recibida y un mensaje de operación incorrecta.
P6	Caso 4 – b – 2	<ul style="list-style-type: none"> <li>• Realizar la lectura del código QR con un dispositivo móvil</li> <li>• Introducir identificador de usuario con mayoría de edad</li> <li>• Introducir contraseña de usuario</li> <li>• Aceptar condiciones del servicio</li> </ul>	El sistema de mostrar los datos de la cuenta de usuario recibida y un mensaje de operación correcta.

Figura 9: Tabla de casos de prueba planteados en el escenario

Según queda definido en el cuadrante y como muestran los casos de uso, P6 resulta ser el caso de éxito y el resto resultan ser fallos que el sistema contempla y deben ser asumidos de forma normal. Es por esto que P6 debe recibir una clasificación por criticidad de alta y debe ser la primera prueba a realizar.

El caso de prueba P5 es la segunda prueba en prioridad, y dado que deben resolver de forma apropiada el dato de mayoría de edad para presentar el resultado, recibe también una criticidad alta.

El caso P4 pretende verificar que el sistema de seguridad a través de la evaluación del código de operación sea correcto. Esto se consigue realizando una prueba P5 o P6 y realizando de nuevo la operación desde el cliente con la misma información, ya que disponemos de los datos de navegación. Se colocaría en tercer orden y su criticidad podría ser alta o media.

Las pruebas P1, P2 y P3, guardan más relación en sí con las pruebas de sistema, ya que verifican poco más que la realización de la lectura del código QR, sin embargo, su verdadero propósito es comprobar que tras cancelar una operación, en cualquiera de sus puntos, el



sistema queda preparado correctamente para realizar una prueba P4, P5 o P6. Podemos asumir que su criticidad es baja, y su ordenación, según quedan especificadas, por conveniencia.

Caso de prueba	Criticidad	Prioridad
P1	Baja	3
P2	Baja	4
P3	Baja	5
P4	Media – Alta	2
P5	Alta	1
P6	Alta	0

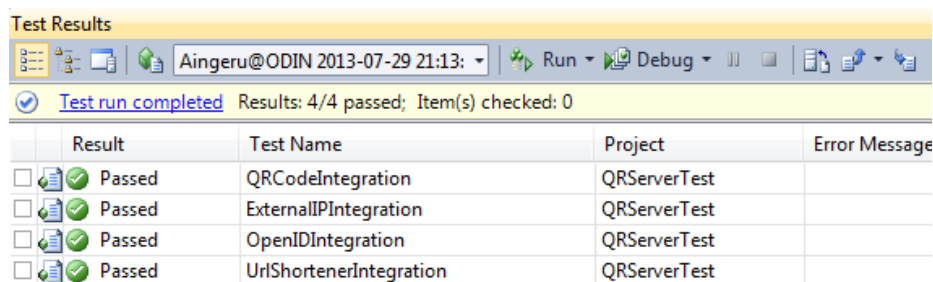
Figura 10: Tabla de categorización de los casos de prueba

Los cuadrantes presentados sobre los casos de prueba nos sirven como hoja de ruta para comprobar, al menos manualmente, que el sistema es funcionalmente correcto. Para comprobar el funcionamiento, solo es necesario aplicar las pruebas por el orden especificado, realizando las tareas indicadas y cotejando el resultado obtenido.

## 5.4.Conclusiones

Los planes de pruebas automatizadas y las herramientas de diseño y desarrollo de pruebas permiten determinar el correcto funcionamiento de un sistema software de forma automática. Disponer de ellas nos permite por tanto conocer el estado o vitalidad del sistema de una forma eficaz y sencilla, sin necesidad de realizar tareas manuales de comprobación. De hecho una práctica habitual consiste en realizar una programación periódica o eventual de las pruebas, para que estas se ejecuten cada cierto tiempo o siempre que se acepte un cambio. Esto multiplica la capacidad de un equipo a la hora de comprobar el funcionamiento de un sistema y no es de extrañar que haya adquirido mucho protagonismo desde hace algún tiempo, proponiendo patrones de desarrollo basados en pruebas como TDD.

El primer paso antes de realizar ninguna tarea de desarrollo sobre el sistema, la ejecución de las pruebas de integración nos indicará si los componentes que consumen los servicios de los que dependemos, funcionan correctamente. De no ser así, antes de nada habrá que averiguar el por qué, puesto que de entrada, nuestro sistema no funcionará correctamente. De hecho conoceremos exactamente en qué servicio se encuentra el problema y aún más, no será necesario utilizar el sistema para depurar el fallo, desde la propia prueba encontraremos el problema y la solución de una forma mucho más sencilla, eficaz y directa.



The screenshot shows a 'Test Results' window with a toolbar at the top containing icons for Test Explorer, Test Results, Run, Debug, and other test-related functions. Below the toolbar, a status bar indicates 'Test run completed' and 'Results: 4/4 passed; Item(s) checked: 0'. The main area contains a table with the following data:

	Result	Test Name	Project	Error Message
<input type="checkbox"/>	Passed	QRCodeIntegration	QRServerTest	
<input type="checkbox"/>	Passed	ExternalIPIntegration	QRServerTest	
<input type="checkbox"/>	Passed	OpenIDIntegration	QRServerTest	
<input type="checkbox"/>	Passed	UrlShortenerIntegration	QRServerTest	

Figura 11: Resultados de las pruebas de integración en el entorno de desarrollo

Si bien es cierto que los servicios expuestos en internet por empresas, sobre todo como Google, no tienden a sufrir variaciones o falta de disponibilidad, tampoco conviene bajar la guardia. Por ejemplo, Los servidores de MyOpenID no han estado disponibles durante algunos días de forma intermitente. No disponer de una visión clara del estado del servicio MyOpenID puede acarrear una gran pérdida de tiempo hasta descubrir que es un fallo ajeno a nuestro sistema. De forma similar, quien iba a pensar que el servicio de descubrimiento de IP externa WhatIsMyIP ampliamente conocido, dejaría de ser de libre acceso. Gracias a la prueba automatizada, fue muy sencillo detectar y tomar una decisión con la que resolver el problema.

Habiendo verificado el correcto funcionamiento de los servicios externos, el siguiente paso es diseñar una serie de pruebas automáticas para verificar el proceso funcional del sistema. Sin embargo las pruebas diseñadas para los servicios cubren un porcentaje muy alto de la funcionalidad del sistema. Podríamos decir que si las pruebas sobre los servicios son correctas, la aplicación de servidor funcionará correctamente con bastante seguridad. Esto unido a que el diseño de pruebas funcionales sobre el sistema es laborioso, nos da como resultado una gran inversión de tiempo para un resultado demasiado pequeño. Además, entendemos que todo el funcionamiento de parte del cliente obedece a sistemas y aplicaciones que quedan fuera de nuestro alcance. Por todo esto se estima que la prueba completa del sistema quede a expensas de una comprobación manual.

Sobre las pruebas de sistema especificadas, solo cabe señalar que han sido llevadas a cabo con éxito. Las aplicaciones de lectura de códigos QR son muy populares y cuentan con equipos especializados de desarrollo y soporte encargados de su correcto funcionamiento, y su disponibilidad para todas las plataformas de dispositivo móvil posibles.

Por último, en cuanto a casos de pruebas, mencionar que en múltiples ocasiones se han utilizado las tablas propuestas para verificar el correcto funcionamiento del sistema. A continuación se muestran dos resultados producidos por la aplicación en servidor, tras procesar las repuestas recibidas, por orden P6 y P4.



Figura 12: Resultado correcto en la aplicación en tiempo de ejecución



Figura 13: Resultado incorrecto en la aplicación en tiempo de ejecución

# Capítulo 6

---

## 6. Historia del proyecto

En este capítulo se narra la historia del propio proyecto desde el punto de vista del alumno. Tanto la situación personal, como la historia del proyecto y las valoraciones del alumno, son expresadas en primera persona para conseguir una perspectiva diferente de los trabajos realizados y sobre todo comprender el contexto en el que se han desarrollado.

### 6.1.Contexto del alumno

En primer lugar deseo hacer una pequeña mención, a como mi trayectoria académica me ha transportado a la situación profesional en la que me encuentro. Desde joven fui un alumno aventajado, aunque a medida de que avanzaba en categoría de estudios iba quedando a la sombra de compañeros mejor estudiantes que yo, hasta llegar a la Universidad Carlos III de Madrid en la que me incluyo en el grupo del estudiante medio, el cual sufriendo ciertas dificultades consigue casi completar el tercer curso de Ingeniería Telemática en algo menos de cuatro años. Acto seguido entro en el perfil típico de alumno que completando todas las asignaturas deja a un lado el Proyecto de fin de Carrera por falta de tiempo y motivación.

Durante el cuarto año académico se me produce una interesante situación en la que para cubrir créditos de libre configuración, decido realizar prácticas en empresa, con la opción de realizarlos en una pequeña y joven empresa de desarrollo de Software, en la que además podría coincidir con antiguos conocidos. Tras completar con mucho éxito las horas estipuladas e intentar volver al mundo académico con una muy recomendable experiencia, la propia empresa me plantea una oferta formal de trabajo que acepté con gusto.

Dicha empresa constituida en aquel entonces por tres jóvenes socios, dos programadores y algún que otro becario, inexperta pero con potencial, me permitió involucrarme profundamente en todos sus proyectos técnicos de los cuales he adquirido muchas y muy diversas experiencias, hasta convertirme irremediabilmente en un miembro clave sobre el que se sustenta la mayor parte del conocimiento técnico. Actualmente y desde hace unos años soy socio activo, mis capacidades como profesional, no tan limitadas como las académicas, mi actitud y sobre todo la pasión por mi trabajo, han hecho de mí el profesional que soy ahora, orgulloso de todo cuanto he conseguido profesionalmente y sufridor de muchas más responsabilidades de las que esperaba.

Esta empresa se dedica hoy en día al diseño y desarrollo de soluciones de identificación automática, siendo especialistas en sistemas de movilidad en el sector profesional.

Actualmente contamos con una cartera importante de clientes en el sector de la lectura de contadores y buenas expectativas de futuro teniendo en cuenta los tiempos que corren.

Como era de esperar una jornada de ocho horas de trabajo no es lo más apropiado para sentarse a desarrollar un proyecto de fin de carrera, peor aún, ser responsable de una de las tres patas de una empresa, hace que una persona como yo quede absorbida más de diez horas al día y que los fines de semana no los utilice para descansar, sino para pensar formas de optimizar el departamento del cual soy responsable. Pero finalmente he reunido las fuerzas suficientes.

## 6.2.Historia del proyecto

Estaba advertido desde el principio en cuanto a las dificultades que iba a tener para poderle dedicar tiempo, y hasta donde alcanza mi memoria, todo aquel que me conoce bien me ha forzado de una manera u otra a no abandonarlo. En dos ocasiones años atrás, comencé el desarrollo del proyecto, pero ambos cayeron en el abandono por falta de tiempo, poca motivación y nula atención por parte del anterior tutor. Fue entonces cuando un buen compañero mío de Ingeniería de Telecomunicaciones terminó su proyecto y me animó a que buscara algún proyecto relacionado con mis conocimientos y mi trabajo, algo que me permitiera valarme de mis experiencias para terminarlo de forma sencilla y limpia.

Impulsado por una pequeña chispa de moral, y quizás algo de envidia, me encontré con un listado algo antiguo de proyectos de fin de carrera relacionados con aplicaciones y sistemas basados en la plataforma Windows Mobile y en el mismo departamento de Telemática. En este momento vi una oportunidad mucho más real de poder sacarlo adelante y no caer en el abandono como en ocasiones anteriores, Teniendo en cuenta que los proyectos que se anunciaban estaban basados en conceptos con los que trabajo día a día y de los cuales soy buen conocedor.

Tardé algún tiempo en conseguir concertar una cita con Daniel, el profesor que figuraba en los proyectos del tablón y mi actual tutor, pero al final mereció la pena. Le expliqué como había llegado hasta él y que tipo de proyecto buscaba, similar a los publicados en el tablón. Desgraciadamente el tablón estaba bastante desactualizado, pero Daniel consiguió en poco tiempo algunas propuestas que involucraban bastante bien los principios que yo buscaba, y rápidamente acepte este proyecto, pensando que a pesar de ser una obligación, me mantendría entretenido.

Aprovechando el tirón de energía extra para afrontar el proyecto, me lancé para dar los primeros pasos. Con los esbozos y las ideas fundamentales del escenario que me planteó Daniel estuve al menos una semana intentando encajar todas las piezas, de forma que todo tuviese el mayor sentido posible. Entre muchas de las ideas que barajé, se encontraban la implementación de un pequeño servidor Web en plataforma Mobile para encajar la

funcionalidad del servidor de códigos QR o proyectos que me sirvieran de inicio para implementar la captura de imágenes para descodificar códigos QR. Todas estas alocadas e innecesarias ideas quedaron atrás en cuanto empecé a descubrir que existía la posibilidad de utilizar las típicas aplicaciones de códigos BIDI. A partir de aquí, comienza una batalla diaria y más o menos constante con las dificultades técnicas que se iban planteando, curiosamente de forma bastante ordenada. En algo más de un mes, conseguí llegar a la implementación del escenario más básico y en el siguiente mes, a ritmo más relajado, llego a un escenario mucho más pulido, incluyendo mejoras en la calidad del Software que implementa dicho escenario.

Tras exponer el desarrollo a Daniel y comprobar que cumple bastante bien el objetivo que perseguíamos, me encuentro en disposición de rematar con el desarrollo de la memoria del proyecto y su posterior presentación. Sin embargo una combinación de problemas personales y profesionales, la llegada de las fiestas de Navidad, las vacaciones de Daniel y una importante falta de motivación para el desarrollo de la memoria, que crecía día a día, hizo que la presentación del proyecto se retrasara alrededor de diez meses más de lo esperado. Una vez superada la combinación mortal de factores que han retrasado el proyecto, consigo terminar de escribir estas líneas y respirar un poco más tranquilo.

### 6.3.Opinión personal

Quizás el momento más crítico para que el desarrollo se haya realizado con éxito, se produce en el momento en el que se define y concreta bien el escenario, partiendo de un Smartphone con una aplicación de códigos BIDI hasta llegar al proveedor de identidad, concentrando los esfuerzos en la idea de establecer una vía de comunicación entre ambos a través de un enlace Web codificado. Todo ello ha conducido a la implementación de una aplicación pequeña pero compleja, que juega con distintos conceptos para cumplir el objetivo que buscábamos. Resaltar también que he puesto todo mi empeño y mis conocimientos para que dicha aplicación cumpla criterios más o menos profesionales de calidad, de forma que los componentes visuales son reutilizables, y los parámetros configurables, bien estructurada y como casi todos mis trabajos orientados a que otras personas puedan realizar un mantenimiento apropiado de ella o algún tipo de evolución.

Otro de los temas más importantes de este trabajo es esta misma memoria, la cual me ha supuesto el mismo tiempo, o quizás más, que en desarrollar del proyecto, lo cual en mi opinión va en contra de las leyes de la documentación. Desde el punto de vista de mi trabajo diario en una pequeña empresa, utilizar el precioso recurso que es el tiempo para documentar tan exhaustivamente, bien debe merecerlo, de lo contrario estaríamos hablando de mucho tiempo perdido, y dicho sea de paso, un trabajo poco interesante para el espíritu técnico.

Si se puede sacar una enseñanza importante de mi historia personal, es sin duda alguna, que es tremendamente difícil compaginar la vida profesional con la académica, y quizás en mi caso particular de forma muy extrema. Durante todos estos años, no haber tenido el título

universitario con todo el esfuerzo y el sacrificio realizado años atrás, es una espina clavada demasiado grande, y quitarla ha costado mucho más de lo que pensaba después de iniciar mi carrera profesional. Se produce un fenómeno bastante común compartido con compañeros que se han encontrado en mi misma situación, y es que al no haber una fuerza imperiosa que te obligue a presentar el proyecto, la fuerza que se dispone para realizarlo es inversamente proporcional, y por tanto se entra en un estado de dejadez del cual es difícil salir.

En un plano más académico, el desarrollo de este proyecto no solo me he permitido comprobar mi experiencia como Ingeniero y poner a prueba mis conocimientos, si no que me ha aportado ideas y conceptos poco explorados para mí, lo cual me enriquece como profesional y le servirán a mi empresa para acceder a parcelas menos habituales. Todo esto me hace estar contento con el trabajo realizado.

# Capítulo 7

---

## 7. Conclusiones

Una vez finalizado el desarrollo del estudio y la implementación del escenario propuesto, solo queda analizar los resultados, evaluar sus consecuencias directas y proponer nuevas líneas de investigación que amplíen el trabajo realizado.

### 7.1. Análisis de resultados

Este estudio ha planteado un escenario en el que se integra el acceso a un sitio Web mediante códigos QR y autenticación descentralizada utilizando servicios OpenID. Este escenario toma la forma de una máquina de Vending que ofrece productos solo a personas mayores de edad. Para acceder al producto la persona en cuestión utiliza su Smartphone, con el cual realiza la lectura de un código QR presentado por la propia máquina. A continuación se desencadenan una serie de peticiones y respuestas entre la máquina, el dispositivo móvil y una tercera parte, encargada de autenticar a dicha persona. Tras resolverse esta conversación la máquina está en disposición de autorizar a la persona para proporcionarle el producto.

Este estudio plantea dos puntos interesantes, el primero es que el usuario puede acceder al producto de una forma extremadamente cómoda, es decir, la lógica de acceso al producto no se encuentra en la máquina. El usuario realiza la lectura del código de producto que desea, sin necesidad de que la máquina disponga de un teclado o interfaz más allá de la presentación del código QR. El segundo punto de interés, es precisamente la autenticación descentralizada, la máquina no realiza ninguna operación de autenticación, ni sería necesario que el fabricante de la máquina disponga de una infraestructura de cuentas de usuario, puesto que nos valemos de un proveedor existente de un servicio estándar y gratuito. Todo esto hace abaratar el coste de la máquina, tan solo valiéndose de tecnologías ya existentes. Y el mayor éxito del estudio radica en este hecho, puesto que el escenario se vale de que se pueda desarrollar con las plataformas hardware y software existentes, sin necesidad de introducir ningún elemento extraño en la vida cotidiana de las personas.

En el plano técnico, el escenario ha sido implementado utilizando tecnologías .Net. Esto es debido principalmente a las aptitudes y experiencia profesional del alumno en dicha tecnología. Si bien es cierto que las tecnologías proporcionadas por Microsoft, raramente son gratuitas, lo que no ocurre con tecnologías Open Source como Java, tienen una tendencia a ser más potentes y eficaces. El hecho de que mediante un solo paquete Software se puedan realizar aplicaciones para cualquier plataforma, así como la integración de módulos nativos de pruebas, control de código fuente, gestión de tareas, compilación continua, y un largo etc.



hacen en general de Visual Studio .Net, un entorno de desarrollo más productivo que sus competidores.

La aplicación implementada para la máquina servidora ha sido diseñada de forma que pueda ser reutilizada más allá de su aplicación en este escenario. Los componentes de codificación QR, comunicación con servicios OpenID y la integración con servicios Google, pueden ser reutilizados con otros propósitos de una forma estándar y sencilla. Esta tarea no es fácil si no se tiene en cuenta desde el principio de la arquitectura de la aplicación. El caso más obvio es el control de códigos QR, basta con referenciar el ensamblado que lo contiene para utilizarlo de forma nativa en la aplicación que se desee.

El plan de pruebas desarrollado nos permite conocer en todo momento el correcto funcionamiento de nuestro sistema. Las pruebas de integración automáticas nos aseguran que los distintos módulos que componen la aplicación funcionan correctamente. Mientras que el cuadro de casos de prueba nos indica que pasos debemos seguir para asegurar el funcionamiento del sistema en todos sus estados. De esta forma, podemos corregir o evolucionar nuestro escenario, con la seguridad necesaria. Efectuando cualquier cambio y ejecutando las pruebas a continuación, averiguaremos si el funcionamiento sigue siendo el correcto, y en caso de no serlo, efectuar las correcciones necesarias y repetir el proceso. De esta forma será muy difícil presentar una implementación que no funcione correctamente.

## 7.2. Riesgos del sistema

Ahora analizaremos los riesgos o puntos débiles de nuestro sistema, así como posibles soluciones. Estos puntos escapan al estudio realizado, puesto que no son el objetivo del mismo.

### 1. Comunicaciones

Gran parte del escenario se desenvuelve en el entorno de comunicaciones TCP/IP. Tanto las conexiones realizadas por el dispositivo móvil hacia el sitio Web de la máquina y al proveedor de identidad, como las realizadas por la máquina servidora, se realizan en claro sin ningún tipo de cifrado. Esto quiere decir que está sujeto a los riesgos de seguridad habituales de este entorno, como son, ataques de denegación de servicio, escudriñamiento de información, envío de peticiones manipuladas, etc.

De la misma forma que aparecen estos riesgos de seguridad, pueden desaparecer con una correcta configuración del servidor Web, que en nuestro caso es la aplicación implementada. La utilización del protocolo HTTPS y establecer unos parámetros de seguridad mínimos evitaría gran parte de los problemas especificados. Normalmente estos parámetros se configuran de una forma sencilla en los servidores Web como IIS o Apache, sin embargo en nuestro caso esto es algo más complicado, puesto que es la propia aplicación la que realiza la función de servidor. Esto nos obligaría a implementar las medidas manualmente. Otra alternativa, o

solución conjunta sería incluir soluciones de seguridad basadas en hardware como por ejemplo cortafuegos.

## *2. Cuentas centralizadas*

Una de las principales críticas aplicables a los servicios de autenticación federada, consiste precisamente en el hecho de que las cuentas de usuario se encuentren bajo la misma máquina, red de máquinas o sistema. Esto implica que un fallo de seguridad en el mismo puede incurrir en una cantidad importante de cuentas de usuario potencialmente usurpadas.

No existe un buen remedio para este riesgo. Lo único que cabe tener en cuenta en este aspecto, es utilizar un proveedor verdaderamente fiable, con medidas de seguridad certificadas y que ofrezca garantías sobre el servicio que ofrece. Lo cual puede ser difícil si se trata de un servicio gratuito.

## *3. Máquina en línea*

El escenario plantea un hecho que no es habitual hoy en día, y es que la máquina de Vending debe disponer de una conexión a Internet para poder operar. Cada vez más son las empresas y usuarios que dependen de servicios en línea, con los llamados SaaS o Software as Service, es imprescindible disponer de una conexión a Internet para poder consumir dichos servicios.

En ningún momento se ha especificado este punto en el escenario, pero no resulta demasiado complicado. Caben dos posibilidades, la primera y más viable es que la máquina disponga de acceso a Internet mediante una tarjeta de datos 3G, de forma que pueda colocarse igual que una máquina convencional. La segunda alternativa viene dada en los casos en los que la máquina no tenga cobertura 3G o GPRS suficiente para poder operar, necesitando acceso por cable, lo cual es más difícil de obtener en según qué casos.

## *4. Dispositivo móvil*

El escenario plantea el acceso al producto mediante la lectura de códigos QR con un dispositivo móvil o Smartphone. Si bien es cierto que es difícil encontrar a alguien sin tal elemento, todavía quedan personas que no disponen de uno, no disponen de una aplicación de lectura en el momento, no saben que hay que hacer con el código, no se fían de él o sencillamente no saben lo que es un código QR. Todo esto lleva a una pérdida potencial de clientes para la máquina.

Bastaría con implementar una emulación del acceso proporcionado por la URL del código QR. De hecho, la aplicación dispone de él, solo hay que pinchar sobre el código QR y se nos dirigirá a un navegador con el acceso en el proveedor de identidad. Sería necesario también especificar unas sencillas e ilustrativas instrucciones a modo de manual de usuario.

## **7.3. Conclusiones generales**

El propósito de este estudio era en esencia conseguir implementar un escenario integrando un servicio OpenID mediante acceso por código QR. Las dificultades técnicas que han aparecido, se han resuelto satisfactoriamente y el sistema diseñado se comporta según el planteamiento propuesto en el estudio. Podemos concluir por tanto que el resultado general es satisfactorio.

El sistema diseñado resulta muy interesante desde el punto de vista técnico. Involucra diversos conceptos, estándares y tecnologías, lo cual lo hace entretenido para técnicos con espíritu investigador y autodidacta. La satisfacción que produce encontrar el sentido a todas las piezas de un puzzle y unir las dentro de un contexto de forma correcta, rara vez se encuentra. En particular resulta de especial interés la integración con servicios Google, los cuales gozan de gran popularidad. En particular en este escenario resulta de especial interés, la forma en la que se integran servicios del mundo Web dentro de un contexto local o de escritorio, ajeno a la plataforma Web.

Cabe destacar la aplicación de herramientas y técnicas de programación avanzadas, como son pruebas automatizadas y creación de componentes, que exigen de un buen conocimiento tanto de la tecnología como de la ciencia, que permiten su aplicación de forma correcta.

Por último solamente destacar nuevamente el éxito del estudio, al cumplir el objetivo fundamental, la implementación de un escenario integrando el acceso por código QR a un servicio con autenticación OpenID.

## 7.4.Trabajo futuro

En cuanto a la ampliación de este estudio, la principal fuente consistiría en cubrir los riesgos propuestos. En mayor o menor medida son riesgos que deben solventarse antes de poder realizar una implantación real del escenario propuesto. Cualquier empresa o particular desplegando un escenario similar se vería afectado por ataques a través de la red de datos. Esto y otros aspectos sujetos a análisis, deberían controlarse en un entorno profesional.

Una segunda fuente, consiste en una sencilla evaluación sobre la aplicación de la máquina servidor. Actualmente solo se contempla el acceso a un único proveedor de identidad, esto podría mejorarse de una forma muy sencilla parametrizando la URL del proveedor de identidad. Desde la interfaz de la aplicación podría seleccionarse el proveedor con el que se debería operar, y de esta forma componer la petición adecuada. De esta manera dotaríamos a nuestro escenario de mucha más flexibilidad.

Una tercera fuente importante de líneas de investigación consecuentes a este estudio, vendrían de estudiar posibles aplicaciones o escenarios alternativos en los que se podría aprovechar el desarrollo realizado. Cada caso requeriría de un estudio pormenorizado, un análisis más detallado y resultados más concretos, encaminados a su aplicación inmediata. El servicio OpenID es en esencia consumible por cualquier sistema que controle cuentas de

usuario, sin embargo la codificación QR es susceptible de utilizarse en entornos poco explorados y menos aún en conjunción con OpenID o simplemente con el concepto de cuentas de usuario. A continuación se muestran algunos de ellos.

- Añadir un código QR en los carteles promocionales permiten a la gente realizar el “Me Gusta” en las páginas de Facebook.
- Poner códigos QR en la ropa para llevar a la gente a su página de Twitter, en la que escribir rápidamente.
- En las fotos, poner los códigos QR que conducen a las personas a los perfiles de LinkedIn.
- Añadir un código QR en su tarjeta de negocios que dirige a la gente al sitio en línea, puede contener un vídeo de usted o de su empresa, enlaces a los perfiles de los medios de comunicación social, y a su sitio Web corporativo.
- Añadir códigos QR a las distintas secciones de un libro que la gente vea directamente aumentar el contenido en la Web registrando el contenido consultado.
- En los restaurantes, poner los códigos QR en los tableros. Estos pueden dirigir a las personas a las páginas de los medios de comunicación social del restaurante.
- Añadir un código QR para un anuncio impreso para un restaurante. Al escanear el código, la gente va de forma directa a una página donde se pueden realizar reservas, también para otros tipos de servicios que requieren de reserva.
- Que sea fácil para la gente comprar, poner un código QR en una pieza de correo directo para un minorista. Llevar a la gente a un sitio optimizado para móviles que les permita buscar y comprar.
- Los códigos QR pueden ser escaneados desde un televisor. Registrando lo que ve en ese momento, o publicando automáticamente en las redes sociales.
- ¡Pagar por cosas! Los códigos QR pueden ser dirigidos a sitios que deducirán automáticamente el dinero de su cuenta por una compra.
- Que sea fácil para la gente ensamblar sus productos. Poner un código QR en la caja o en las instrucciones que les permiten ver vídeos que muestran cómo armar algo, registrando lo que se ha armado.

- Que sea fácil para las personas encontrar su negocio. Poner un código QR en uno de sus anuncios impresos que conecte a las personas a Google Maps.
- Dar a la gente algo que hacer mientras esperan. Si a menudo tiene un público cautivo que está esperando en línea, poner un código QR en un cartel o valla publicitaria que esté cerca. Esto les permitirá interactuar con su marca, mientras que ellos están esperando.
- Donaciones para recaudar fondos. Que sea lo más fácil posible para que la gente aporte, por la adición de un código QR que los lleve a una página de contribución.
- Juegos como caza del Tesoro. Por un poco de diversión, añadir los códigos QR a lo largo de una cierta área, para acceder a sus juegos preferidos.
- Poner un código QR en la parte inferior de su blog. Cuando la gente imprima y guarde los mismos, van a ser capaces de obtener fácilmente su blog en el futuro.
- Haz una pregunta del trivial. El código QR puede conducir a la gente a una forma simple en la que puede responder para ganar.
- Ayudar a las personas a seleccionar la mejor bebida, añadir un código QR en las etiquetas de botellas de vino que les permiten ver una visita virtual de su bodega y registrar los vinos que se han tomado.
- Que sea fácil para alguien hacer “Check-In” a su negocio. Poner un código QR fuera de su tienda, para la exploración que permita a alguien poder “Check-In” a través de un sitio como Foursquare.
- ¿Está hablando o presentando en una conferencia? Incluir un código QR en sus folletos para que la gente puede aprender sobre usted, su empresa, con contenido multi idioma ajustado personalmente.
- Proporcionar instrucciones adicionales para los ejercicios de terapia física. Además de las medidas de inclusión en el material impreso, permita a las personas obtener más información en línea sobre las aflicciones que padece.
- Permita que las personas interactúen con sus tarjetas de felicitación. Claro, la gente puede querer escuchar una nota de felicitaciones y agradecimiento. Un código QR les permite ver una imagen o vídeo que también se hace eco de los saludos.
- Mejorar su oferta de servicio al cliente. La próxima vez que distribuya notas de la versión, o manuales de capacitación, los códigos QR pueden vincular a las personas a

tutoriales en línea o formas de llegar a su equipo de apoyo a través de su teléfono móvil.

- Cosas para todos los días. Hay cosas que la gente compra todos los días, café, refrescos, agua. Poner códigos QR en las etiquetas de los productos para ayudar a involucrar a sus consumidores más fieles.
- Proporcionar información nutricional en los menús para ayudar a la gente a decidir si algo es bueno para ellos, desde un punto de vista dietético.

Si un usuario dispone de una cuenta OpenID y un Smartphone capaz de leer códigos QR, dispondría de un sinfín de servicios personalizados de forma muy cómoda y automática, como los citados. De hecho, estos solo son algunas de las posibles ideas sobre investigaciones al respecto, con lo que a los servicios accesibles ahora, podríamos añadir nuevos servicios en el futuro sin necesidad de realizar ninguna adaptación, puesto que se accede a través del estándar OpenID.

# Apéndice A

## A. Presupuesto

En este apartado se presentará el presupuesto que hace referente al coste que presenta el esfuerzo para realizar un estudio técnico como el descrito a lo largo de esta memoria. Se detallan los costes de los materiales utilizados así como el coste de personal. Se describe el coste del estudio a nivel material, personal e instalaciones, tomado como si dicho estudio hubiera sido realizado por una única persona.

### 1. Costes de personal

La duración del proyecto ha sido de 3 meses. En cuanto a la dedicación diaria hacia el mismo, se puede considerar una jornada completa de 8 horas, ya que aunque no durante toda la jornada se ha dedicado a la realización de la memoria, si han sido necesarias muchas horas de estudio y análisis previo para completar la misma.

Una dedicación de 8 horas al día por 5 días a la semana durante 3 meses se traduce en un total de 480 horas. En este periodo se incluyen únicamente las horas dedicadas a realizar el estudio y la memoria, no contabilizándose tiempos de trámites, gestión o periodos de inactividad.

En cuanto a los honorarios de un ingeniero, estos responden a acuerdos de libre mercado en base a acuerdos comerciales entre particulares y profesionales. En el pasado, el Colegio Oficial de Ingenieros de Telecomunicaciones publicaba a modo orientativo los honorarios que debía cobrar un ingeniero por hora. Con la nueva normativa europea, los colegios han dejado de publicar dicha información, con lo que para el cómputo del coste humano del proyecto no podemos usar una información actualizada. Tomaremos una referencia anterior para dicho cálculo en la que se estipulaba un coste de 75 euros por hora de ingeniero. Por otra parte, el coste de un director de proyecto se estima en un 7% del coste total del proyecto. En este caso existirá un director, contabilizando así el tiempo dedicado por parte del tutor asignado por parte de la universidad.

La siguiente tabla resume los costes debido a los honorarios de las partes descritas.

Concepto	Coste	Cantidad	Total
Ingeniero de Proyecto	75 € / Hora	480 horas	36.000 €
Director de Proyecto	7%	36.000 €	2.520 €

<b>Total sin IVA</b>	<b>38.250 €</b>
<b>Total con IVA (21 %)</b>	<b>46.282,5 €</b>

## 2. Costes de material

El coste del material que se expone en esta sección se refiere al material usado para el estudio y desarrollo de la presente memoria así como las instalaciones utilizadas para la misma.

### 1. Entorno de Trabajo

El grueso del proyecto se ha realizado en nuestro despacho alquilado, usando las dependencias e instalaciones de la empresa. El coste del despacho contratado con el centro de negocios *Atrium* es de 390 € mensuales, incluye los siguientes conceptos.

- Despachos amueblados según sus necesidades
- Atención personalizada de las llamadas telefónicas con el nombre de su empresa
- Recepción de correspondencia y paquetería
- Atención de visitas
- Recepción / envío de faxes
- Acceso a salas de reuniones con medios audiovisuales
- Conexión a Internet de banda ancha (también acceso WIFI)
- Soporte técnico y asesoramiento informático
- Línea telefónica sin recargo
- Domiciliación fiscal, legal y comercial
- Acceso al centro de negocios 24 horas / 365 días
- Servicios de impresión, fotocopidora y escáner
- Cuenta de correo electrónico
- Limpieza diaria
- Aire acondicionado y calefacción
- Prensa diaria en zonas comunes
- Seguridad 24 horas
- Horario de atención continuado de 9:00h a 20:00h (agosto de 9:00h a 17:00h)

Esto supone incluir, en concepto de entorno de trabajo, dentro de los costes del proyecto la cantidad de 1.170 euros que hace referencia al periodo de 3 meses de duración del proyecto.

### 2. Equipo Informático

El equipamiento informático necesario para el desarrollo del estudio plasmado en esta memoria tiene un precio de mercado de unos 1.372,9 euros. Dentro del mismo se incluyen los elementos descritos a continuación.



Concepto	Coste
Material Informático: <ul style="list-style-type: none"> <li>Sony Vaio SVE1513C1EW P2020/4GB/500GB/HD 7650/15.5"/Blanco</li> <li>Alfombrilla Ergonómica Azul Gel</li> <li>Logitech B110 Optical USB Mouse Negro OEM</li> </ul>	458.9 €
Dispositivos móviles: <ul style="list-style-type: none"> <li>iPhone 4S 16Gb</li> <li>Sony Xperia U</li> <li>BlackBerry Curve 3G 9300</li> </ul>	893 €
Tarjeta SIM Movistar prepago 1GB+50 min	9 € /mes
<b>Total Proyecto</b>	<b>1.372.9 €</b>

La depreciación que sufren los sistemas y programas informáticos según el Real decreto 1777 de 2004 es de un 33 % por año. Por tanto, el valor residual del equipo después de 3 meses de ejercicio es de un 8,25%. Se incluirán por tanto 111.53 euros en los costes materiales más 21 € de tarjeta SIM dentro del concepto de equipo informático, lo que hace un total de 132,53 euros.

### 3. Licencias Software

El sistema operativo instalado en el equipo informático en el que se ha realizado el proyecto es Windows 7 Professional de Microsoft. Además es necesario otros paquetes de software: Visual Studio Professional, Team Foundation Server, Microsoft Office 2010. El coste de este software es de 2298 euros al año. La depreciación que sufren los sistemas y programas informáticos según el Real decreto 1777 de 2004 es de un 33 % por año. Por tanto, el valor residual del aparato después de 3 meses de ejercicio es de un 8,25%. Se incluirán por tanto 189,58 euros en los costes del proyecto en concepto de licencias de Software.

### 4. Material de Oficina

En cuanto al material de oficina usado durante el proyecto, se asume que su coste va implícito en el coste del alquiler del despacho en concepto de *Entorno de Trabajo*.

Concepto	Coste	Cantidad	Total
Entorno de Trabajo	390 € / mes	3 meses	1.170 €
Equipo Informático	1.351.9 €	8,25 % depreciación	132,53 €
Licencias Software	2.298 €	8,25 % depreciación	189,58 €
Material de Oficina	-	-	-

<b>Total sin IVA</b>	<b>1.492.11 €</b>
<b>Total con IVA (21%)</b>	<b>1.805,45 €</b>

### 3. Presupuesto completo

Teniendo en cuenta lo descrito anteriormente, el precio total para el proyecto completo se resume en la siguiente tabla.

Concepto	Coste
Total Material	1.805.45 €
Total Honorarios	46.282,5 €
<b>Total Proyecto</b>	<b>48.087,95 €</b>

# Apéndice B

---

## B. Glosario de términos

En este apéndice se detallan gran parte de los términos menos mundanos utilizados a lo largo de la memoria. Estas definiciones pueden ayudar a comprender mejor las explicaciones y los desarrollos realizados.

- 3G, abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles)
- Apple, empresa multinacional estadounidense con sede en Cupertino, California, que diseña y produce equipos electrónicos y software. Entre ellos los más conocidos son iPod, iPhone y MaciOS.
- Aplicación de lectura de códigos. Software destinado generalmente a dispositivos móviles para la lectura de códigos de barras, sin la cual, estos dispositivos no tienen la capacidad de interpretarlos.
- Arquitectura Single Sign On, o SSO, es un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación. Se podría definir también como sistema centralizado de autenticación y autorización.
- Android, sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil, desarrollados por Android Inc., que Google respaldó económicamente y más tarde compró en 2005.
- Autenticación, acto de establecimiento o confirmación de algo o alguien como auténtico. En el contexto de las TIC, proceso de intento de verificar la identidad digital del remitente de una comunicación.
- Autenticación descentralizada, proceso de autenticación llevado a cabo por una tercera parte verificadora. En contraposición a la autenticación centralizada, en la cual el mismo sistema al que se intenta acceder, verifica la identidad.
- Apache, implementación de servidor Web de código abierto.
- API, o Interfaz de programación de aplicaciones, es el conjunto de estructuras y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- Adaptador de red, o NIC, periférico que permite la comunicación con dispositivos conectados entre sí y la compartición de recursos. Existen diferentes tipos según su cableado o arquitectura, el más común actualmente es el Ethernet.

- Blog, también bitácora digital, cuaderno de bitácora, ciberbitácora, ciberdiario, o web log, es un sitio web en el que uno o varios autores publican cronológicamente textos o artículos
- Codificación QR, conjunto de técnicas y procedimientos para la conversión de cierta información en un determinado sistema de datos, que en este caso es un código QR.
- Código QR, o código de respuesta rápida, es un módulo útil para almacenar información en una matriz de puntos o un código de barras bidimensional creado por la compañía japonesa Denso Wave, subsidiaria de Toyota, en 1994
- Codificación bidimensional, en contraposición a las codificaciones unidimensionales que contienen información decodificable con un haz lineal, las codificaciones bidimensionales albergan información trazada en un plano.
- Chart API, API de libre acceso diseñado por Google para la visualización de información en forma de gráficos y esquemas.
- Caso de uso, descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.
- Cortafuegos, dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios
- Cuentas centralizadas, el hecho de que las cuentas de usuario de un determinado sistema informático, se encuentren gestionadas por el mismo.
- Caso de prueba, conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.
- Check-In, proceso mediante el cual un recepcionista registra la llegada de un cliente a un hotel, aeropuerto o puerto. Recientemente, el término se ha extendido también a algunas redes sociales que, permiten a un usuario comunicar dónde se encuentra en un momento determinado al resto de usuarios de la red.
- Code128, codificación de barras de alta densidad, usado ampliamente para la logística y paquetería.
- Denso Wave, empresa creadora del código QR.
- Dispositivo móvil, sistema Hardware diseñado para comunicaciones de voz y datos, portable y con autonomía limitada.
- Dirección IP, etiqueta numérica que identifica, de manera lógica y jerárquica, a un interfaz de un dispositivo dentro de una red que utilice el protocolo IP. Existe una diferenciación básica entre direcciones privadas o internas y direcciones públicas o externas, de forma que desde una dirección pública no se puede acceder a una privada.
- DynDNS, compañía de Internet de los Estados Unidos de América dedicada a soluciones de DNS en direcciones IP dinámicas
- Desafío, petición enviada por un sistema verificador para que la entidad cliente devuelva su resolución.

- Depuración, conjunto de técnicas, herramientas y procedimientos para la detección y corrección de errores de programación en sistemas Software.
- Dependencias externas, son sistemas de terceros de los cuales depende directamente el correcto funcionamiento del nuestro.
- Denegación de servicio, DoS, es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente producido por un sobre exceso de peticiones que cause el malfuncionamiento del equipo servidor.
- Dirasoft, empresa dedicada al desarrollo de Software, venta de dispositivos de identificación automática e integración de soluciones tecnológicas para empresas de todo tipo.
- Expresión regular, o Regex, es una secuencia de caracteres que forma un patrón, principalmente utilizado para la búsqueda de patrones de información u operaciones de sustituciones.
- ERP, Enterprise Resource Planning, es un sistema Software encargado de planificar y gestionar los recursos de una corporación. Asumiendo que existen características comunes entre las empresas independientemente de su negocio, existen paquetes ERP estándares que administran los recursos de empresas de forma óptima.
- EAN13, o European Article Number, estándar de codificación de barras gestionado por GS1. El código EAN más usual es EAN13, constituido por 13 dígitos y con una estructura dividida en cuatro partes.
- Ensamblado, en el contexto de .Net, un archivo de librería DLL o ejecutable EXE, contenedores de clases y métodos reutilizables por otros ensamblados.
- Escudriñamiento de información, ataque informático que tiene como objetivo obtener información de una red a la que están conectados los usuarios sin modificarla. La técnica conocida como Sniffing utiliza el adaptador de red local para obtener información que viaja por la red, mientras que el Snooping utiliza otros medios para obtenerla en local.
- Función delegada, en el contexto de .Net, es una función especificada por parámetro a otro procedimiento para que esta sea invocada por él. De esta forma no se realiza la ejecución directa del método, si no que se delega la ejecución a otro método, responsable de su invocación.
- Fraunhofer FOKUS, organización I+D+i dedicada entre otras cosas a las comunicaciones máquina a máquina en su proyecto OpenMTC.
- Formulario .Net, clase nativa que sirve como contenedor principal de controles visuales y toma la forma de una ventana clásica en entornos Windows.
- Foursquare, servicio basado en localización web aplicada a las redes sociales. La geo localización permite localizar un dispositivo fijo o móvil en una ubicación geográfica
- Facebook, sitio web de redes sociales creado por Mark Zuckerberg, orientado a las relaciones personales, amistades, publicación de imágenes, noticias, y aplicaciones de todo tipo.

- GMail, potente e innovador servicio de correo electrónico gratuito proporcionado por la empresa estadounidense Google, Inc. desde el 15 de abril de 2004.
- Google, empresa multinacional estadounidense especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías.
- Goo.gl, servicio de acortamiento de URL proporcionado por Google Inc. De forma gratuita limitada. De esta forma una URL de cierta longitud puede ser transformada en una sustancialmente más corta y resuelta mediante redirección.
- GPRS, servicio general de paquetes vía radio creado en la década de los 80 para la transmisión de datos mediante conmutación de paquetes.
- GPS, o Global Positioning System, sistema que permite determinar en todo el mundo la posición de un objeto, una persona o vehículo con bastante precisión.
- Google Maps, servicio gratuito de Google como aplicación de mapas en la web. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo entero e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle.
- GSM, Global System for Mobile Communications y originalmente Groupe Spécial Mobile, es el estándar desarrollado por el Instituto de Estándares de Telecomunicaciones Europeo o ETSI para describir protocolos de segunda generación o 2G en redes digitales celulares para teléfonos móviles.
- HTML, siglas de Hypertext Markup Language, estándar a cargo de la W3C para la elaboración de páginas Web, y basado en la referenciación mediante enlaces de texto a ubicaciones.
- HTTPS, protocolo seguro de transferencia de hipertexto, es un protocolo de nivel de aplicación basado en HTTP y destinado a la transferencia segura de datos de Hipertexto
- HTTP, Hyper Text Transfer Protocol, protocolo de nivel de aplicación desarrollado por la World Wide Web Consortium y la Internet Engineering Task Force que define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web para comunicarse.
- IBM, empresa multinacional estadounidense fundada en 1911, de tecnología y consultoría con sede en Armonk, Nueva York. Fabrica y comercializa hardware y software para computadoras, ofrece servicios de infraestructura, alojamiento de Internet y consultoría en una amplia gama de áreas relacionadas con la informática.
- iOS, sistema operativo móvil de la empresa Apple Inc., originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en otros dispositivos de la compañía. Se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix.
- interfaz de aplicación, conjunto de elementos visuales que permiten al usuario interactuar con ella. Elementos como ventanas, botones y textos conforman el interfaz de la aplicación.

- infraestructura de cuentas de usuario, elementos Hardware y Software que permiten a una aplicación operar con procedimientos de autenticación y autorización de los usuarios que acceden a ella.
- identificación automática, es el nombre que recibe en el ámbito profesional, el conjunto de tecnologías que permiten el diseño e implementación de dispositivos para el reconocimiento de información sin intervención humana. En concreto los dispositivos de lectura de códigos de barras conforman gran parte de este término.
- IIS, Internet Information Services, implementación de servidor Web de Microsoft.
- JavaScript, lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en el lado del cliente (client side), implementado como porciones de código sobre HTML permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.
- Login, proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por él.
- Lenguaje de programación de alto nivel, es aquel que se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas. Por ejemplo Java, C# o VB.Net
- Lenguaje Java, lenguaje de alto nivel originalmente desarrollado por James Gosling de Sun Microsystems, más tarde adquirida por la compañía Oracle.
- Lenguaje .Net, es aquel perteneciente al conjunto de lenguajes amparados por las tecnologías .Net de Microsoft.
- LinkedIn, red social para profesionales de todos los ámbitos.
- Multiplataforma, se habla de una aplicación multiplataforma cuando su funcionamiento es independiente del dispositivo que la ejecuta. Las aplicaciones interpretadas (como las basadas en Java o .Net) son en esencia multiplataforma, puesto que es el intérprete diseñado para el dispositivo en cuestión el encargado de ejecutarlas.
- Me Gusta, popular opción de la red social Facebook, en la que se puede marcar una fotografía o comentario positivamente para que el resto de usuarios lo conozcan.
- Multi hilo, característica de aplicaciones software complejas, en las que existen más de un sub proceso en ejecución bajo la misma aplicación.
- Microsoft, empresa multinacional de origen estadounidense, fundada el 4 de abril de 1975 por Bill Gates y Paul Allen. Dedicada al sector del software, desarrolla, fabrica, licencia y produce software y equipos electrónicos, siendo sus productos más usados el sistema operativo Microsoft Windows y la suite Microsoft Office.
- MyOpenID, popular proveedor de identidad basado en el estándar OpenID.
- M2M, comunicaciones máquina a máquina.

- NAT, Network Address Translation, es el mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles.
- OpenID, estándar de identificación digital descentralizado, con el que un usuario puede identificarse en una página web a través de una URL y puede ser verificado por cualquier servidor que soporte el protocolo.
- OpenMTC, proyecto desarrollado por la Fraunhofer FOKUS para la implementación de redes M2M de gran escala.
- Proveedor de identidad, entidad con la infraestructura digital necesaria para permitir el protocolo OpenID.
- Plataforma .Net, es el intérprete de aplicaciones .Net permitiendo así su ejecución.
- POST, método especificable para la petición HTTP enviada al servidor. Su uso está relacionado con el envío de información al servidor, en contrapartida al método GET que especifica la petición de un recurso.
- PictureBox, control estándar .Net que permite establecer una imagen en un contenedor, por ejemplo de tipo ventana, para que el usuario la pueda visualizar en el interfaz.
- Pruebas automatizadas, hace referencia a la programación de métodos de prueba que ejecutan en un entorno controlado los métodos de la aplicación a probar. De esta forma se automatiza la ejecución de las pruebas que son necesarias para establecer el correcto funcionamiento de nuestra aplicación.
- Pruebas de integración, son aquellas pruebas destinadas a la verificación del correcto funcionamiento entre los diferentes sistemas físicos o lógicos dentro de nuestra aplicación, como servicios, bases de datos, dispositivos controlados, etc.
- Pruebas funcionales, son aquellas pruebas destinadas a la verificación del correcto funcionamiento de las características finales a disposición del usuario, generalmente a través de su interfaz.
- Pruebas de sistema, son aquellas pruebas destinadas a la verificación del correcto funcionamiento de nuestra aplicación desplegada sobre diferentes dispositivos o plataformas.
- Petición manipulada, aquellas peticiones enviadas a servidor, por ejemplo HTTP, que han sido recogidas por un tercero y alteradas en cierta manera y presumiblemente maliciosa.
- Router, o enrutador, es un dispositivo que proporciona conectividad a nivel de red o nivel tres en el modelo OSI.
- Sistema de movilidad, término manejado en el ámbito profesional para describir aquellos sistemas informáticos dispuestos para operar en campo de forma móvil.
- SaaS, Software as a Service, término para describir las aplicaciones Software que operan desde un entorno independiente del que se encuentra el usuario, ofrecido así



como un servicio del cual no se tiene control. En general las aplicaciones en la nube son SaaS.

- System.Net.TCPListener, Clase .Net desde la cual se pueden controlar fácilmente hilos de escucha a nivel TCP.
- Socket TCP, Clase .Net con la cual manejar comunicaciones TCP a bajo nivel.
- Servicio Web, o Web Service, aplicación Web sin interfaz destinada a ofrecer funciones distribuidas que permiten la integración con las aplicaciones consumidoras.
- StackOverflow, popular foro de discusión y respuestas sobre todo tipo de tecnologías Software.
- Servicio de autenticación propietario, procedimiento de verificación de identidad de usuarios para el acceso a la aplicación de implementación propia y no estándar.
- Smartphone, dispositivo móvil de última generación con el que se pueden realizar todas las funciones de un teléfono y dispone de las capacidades informáticas de un equipo convencional.
- Software, equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.
- SCADA, siglas de Supervisory Control and Data Acquisition, son aquellas aplicaciones destinadas a la recolección de datos en planta de los dispositivos gestionados por la misma así como su control operativo.
- Tiempo de ejecución, intervalo de tiempo en el que un programa informático se ejecuta en un sistema operativo. Este tiempo se inicia con la puesta en memoria principal del programa, tras lo cual este comienza a ejecutar sus instrucciones. El intervalo finaliza en el momento en que éste envía al sistema operativo la señal de terminación.
- TDD, Test Driven Development, técnica de programación que involucra otras dos prácticas, escribir las pruebas primero (Test First Development) y Refactorización (Refactoring). Para escribir las pruebas generalmente se utilizan las pruebas unitarias.
- TCP/IP, dos protocolos importantes en el nivel de red. Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), fueron dos de los primeros en definirse, y los más utilizados de la familia. Permiten el establecimiento de conexiones de datos y la transmisión de paquetes a sus destinatarios, entre otras funciones fundamentales.
- Twitter, particular red social también definido como servicio de microblogging. La red permite enviar mensajes de texto plano de corta longitud, con un máximo de 140 caracteres, llamados tweets, que se muestran en la página principal del usuario y en todos aquellos usuarios interesados en ellos.
- Toyota, una de las tres grandes empresas automovilísticas japonesas, desafiando a los fabricantes de automóviles estadounidenses. Produce automóviles, camiones, autobuses y robots, siendo además es la quinta empresa más grande del mundo.
- TPV, Terminal de puesto de venta, dispositivo móvil para la realización de cobros de forma remota.

- URL, o localizador de recursos uniforme, es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet como imágenes, videos o páginas Web.
- URI, o identificador uniforme de recursos, es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema. Los URI pueden ser localizadores uniformes de recursos (URL), nombre de recurso uniforme (URN), o ambos.
- Visual Studio, IDE o entorno de desarrollo diseñado por Microsoft y destinado a fabricar aplicaciones sobre tecnologías .Net.
- Verificación de identidad descentralizada, procedimiento por el que se realiza la verificación de la identidad de un usuario en un contexto diferente o por una entidad externa a la que ejecuta la aplicación que requiere la verificación.
- Vitalidad del sistema, indicador a niveles generales del correcto funcionamiento de un sistema Software.
- Vending, neologismo en voz inglesa que se utiliza para denominar el sistema de ventas por medio de máquinas auto expendedoras accionadas por diversos medios de pago.
- Windows Mobile, sistema operativo obsoleto y con soporte limitado propiedad de Microsoft, destinado a dispositivos móviles.
- WinForms Designer, herramienta de Visual Studio que permite el diseño de ventanas o formularios Windows visualmente, arrastrando controles y estableciendo propiedades sobre ellos sin necesidad de codificar. El propio diseñador genera el código fuente asociado a interfaz diseñado con todo tipo de ayudas.
- WhatIsMyIP, sitio Web dedicado a facilitar a los usuarios la dirección IP de salida a internet, entre otros servicios de red de utilidad.

# Apéndice C

---

## C. Referencias

En este apéndice se indican las fuentes a las cuales ha sido necesario acudir, para encontrar respuestas a preguntas planteadas durante el estudio, o de las cuales se ha extraído el contenido que conforma alguno de los capítulos.

### 1. Microsoft Developer Network

Portal de acceso para desarrolladores sobre tecnologías .Net de Microsoft. Pone a disposición de suscriptores y visitantes, las herramientas en evaluación, noticias, ejemplos y cursos de aprendizaje, así como toda la documentación oficial disponible sobre el API .Net en sus distintas versiones y plataformas.

<http://msdn.microsoft.com>

### 2. Google Developers

Portal de acceso para desarrolladores sobre productos Google. Pone a disposición de todo el público, noticias, ejemplos y cursos de aprendizaje, así como toda la documentación disponible sobre los productos de Google como Google+ o Google Maps, para su integración en proyectos.

<https://developers.google.com/>

### 3. OpenID

Sitio Web oficial de la fundación OpenID. Ofrece una gran presentación sobre el servicio, así como noticias y acceso a los desarrolladores en el grupo de trabajo OpenID.

<http://openid.net/>

## 4. StackOverFlow

Popular foro sobre tecnologías de Software. Una gran fuente de conocimiento, a la que acuden desarrolladores sobre todo tipo de tecnologías, planteando cuestiones y respuestas sobre todo tipo de ámbitos y niveles.

<http://stackoverflow.com/>

## 5. Wikipedia

Enciclopedia libre y gratuita, construida a través de esfuerzo colaborativo. Fuente inagotable de conocimiento sobre toda materia.

<http://es.wikipedia.org>

## 6. Otros

A continuación se muestran una serie de fuentes menores que han sido de utilidad para el desarrollo del estudio y su documentación.

<http://eduarea.wordpress.com/2011/08/22/100-ideas-acerca-de-como-utilizar-los-codigos-qr/>

<http://dyn.com/dns/>

<http://www.whatismyip.com/>

<http://blogcmt.com>

<http://www.open-mtc.org>